
6,– DM öS 50,– sfr 6,–
CPC
Amstrad
INTERNATIONAL
CPC · PCW JOYCE · PC
1
Januar 1990
6. Jahrgang
CPC-Tophits
– Roboter selbstgebaut
– Alles über RSX-Programmierung
– Supertools für MC-Programmierer
– Neue 1-kByte-Programme
ASSEMBL
LISP
FORT
Entscheidungshilfe
Welche Programmiersprache ist die richtige?
Neuer PC-Teil!
– Einführung in MS-DOS
– Helfer für Batch-Files
– Datensicherung selbstprogrammiert
– Viele tolle Tips & Tricks
– Die neuesten Spiele
und vieles mehr!
PCW
– MS-DOS-Formate auf PCW
– Bildschirmsteuerung ohne BASIC
– JUCA-Adreßverwaltung Teil II
Viele Tips & Tricks für alle Amstrad-Computer

Impressum

Herausgeber
Christian Widuch
Chefredakteur
Stefan Ritter
Stv. Chefredakteur
Michael Ebbrecht (me)
Redaktion
Claus Daschner (cd), Joachim Freiburg (jf)
Redaktions-Assistenz
Anke Kerstan (ke), Susanne Eska (es)
Schlußredaktion
Vera Brinkmann
Produktionsleitung
Gerd Köberich
Bereichsleitung
Christian Heckmann (Fotografie/Layout)
Claudia Küllmer (Fotosatz/Lektorat)
Margarete Schenk, Helmut Skoupy (Montage/Reprografie)
Layout
Michael Grebenstein
Fotografie
Heike Reifer
Fotosatz
Marcus Geppert
Lektorat
Dagmar Wilhelm-Ballhaus
Montage/Reprografie
Dieter Schnobl, Andrea Gundlach
Werbegestaltung
Mohamed Hawa, Petra Küch
Anzeigenverkaufsleitung
Wolfgang Schnell
Anzeigenverkauf für PLZ 1, 4, 5
Gerlinde Rachow, Telefon: (05651) 809390
Wolfgang Brill, Telefon: (05651) 809380
Anzeigenverkauf für PLZ 2 + 3
DMV-Verlagsbüro Hamburg
Ohlsdorfer Straße 34, 2000 Hamburg 60
Sylvia Ehrenpfordt, Telefon: (040) 461233, Telefax: (040) 474310
Anzeigenverkauf für PLZ 6 – 8
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon: (089) 4391087, Telefax: (089) 4391080
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle
Anzeigenverwaltung und Disposition
Andrea Giese, Karina Ehrlich, Beate Kranz
Anzeigenpreise
Es gilt die Anzeigenpreisliste Nr. 4 vom 01. 01. 1989.
Anzeigengrundpreise
1/1 Seite sw DM 5240,–
Farbzuschlag: erste und zweite Zusatzfarbe aus
Europaskala je DM 750,–.
Vierfarbzuschlag DM 2250,–.
Anschrift Verlag/Redaktion:
DMV Daten und Medien Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege
Telefon: (05651) 809–0
Telefax: (05651) 809333
Vertrieb
Verlagsunion Erich Pabel – Arthur Moewig KG (VPM)
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Druck
Druckerei Jungfer, 3420 Herzberg
Bezugspreise
»PC Amstrad International« erscheint monatlich am Ende des Vormonats.
Einzelpreis DM 6,–/sfr. 6,–/öS 50,–
Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.
Inland:
12 Ausgaben: DM 66,–
6 Ausgaben: DM 33,–
Europäisches Ausland:
12 Ausgaben: DM 96,–
6 Ausgaben: DM 48,–
Außereuropäisches Ausland:
12 Ausgaben: DM 120,–
6 Ausgaben: DM 60,–
Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr.: 23043–608
Raiffeisenbank Eschwege:
BLZ: 52260385, Kto.-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.
Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernimmt der Verlag keine Haftung. Die Zustimmung zum Abdruck wird vorausgesetzt.

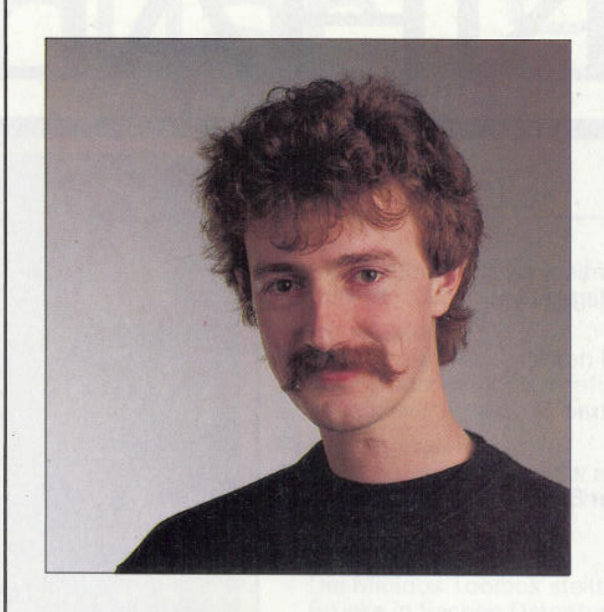
Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.

Amstrad ist das registrierte Warenzeichen der Fa. Amstrad International SA und wird von DMV mit Genehmigung der Fa. Amstrad im Titel dieser Zeitschrift verwendet.

Die Zeitschrift PC Amstrad International ist kein offizielles Organ der Fa. Amstrad und unterliegt völlig der Verantwortung des DMV-Verlages. Der Inhalt der redaktionell von Amstrad gestalteten Seite AMS-Line unterliegt der presserechtlichen Verantwortung der Fa. Amstrad Deutschland GmbH, Dreieichstr. 8, 6082 Mörfelden-Walldorf.

Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW), Bad Godesberg.
ISSN 0935-9095





Teufel und Beelzebub

Viren sind in aller Munde. Jeder fürchtet den Moment der ersten Konfrontation mit diesen nur selten zur Erheiterung beitragenden Programmen. Da rasen im günstigsten Falle die Buchstaben, die sich gerade auf dem Monitor befinden, gen unteren Bildschirmrand. Wer sich einen bösartigen Virus gefangen hat, muß allerdings mit Schlimmerem rechnen. Doch egal, wie bös- oder gutartig eine solche Speichermanipulation auch ist, Ärger bringt sie allemal.

Nun scheint es allerdings, daß trotz allen Unheils, das diese programmierten Getüme anrichten, auch eine winzige positive Ausstrahlung von jenen ungeliebten Gästen auf unser gesellschaftliches Verhalten abstrahlt.

Denken wir nur an den 13. Oktober zurück – einen Freitag! War noch vor einigen Monaten in Zeitungen, Radios und Fernsehen an solchen Daten der Aberglaube das "Thema Nummer eins", so hörte man plötzlich nur noch von den gefürchteten Viren, die das scheinbare Unglücksdatum als Anlaß nahmen, sich selbst zum Leben zu erwecken.

Die Angst vor dem Datenverlust steht also plötzlich in offener Konfrontation zum bis dahin so beliebten Aberglauben. Schwarze Katzen, die in geschlossenen Räumen, in denen jemand einen Schirm öffnet, von links nach rechts unter stehenden Leitern hindurchsausen, gehören somit bald auf die monatlich erscheinende Liste der Dinge, die "out" sind.

Wenn es denn also der Virus an sich und insbesondere die Angst vor diesem am Freitag, den 13ten, schafft, ein Stück Aberglauben von uns zu nehmen, so ist dem Ganzen wenigstens eine gute Seite abzugewinnen, wenn auch der sprichwörtliche Teufel mit dem ebensolchen Beelzebub ausgetrieben wird.

Ein Wort noch zu diesem Heft. Wie Sie sicherlich bald bemerken werden, haben wir den Teil dieser Ausgabe, der sich mit den PCs beschäftigt, um einige Seiten verlängert. Diese Erweiterung ist notwendig, da sich die Palette der Amstrad-PCs durch die 20er-Reihe wesentlich vergrößert hat, und wir auch in diesem Punkt nicht mit Informationen geizen wollen.

Das heißt natürlich nicht, daß wir die Besitzer des CPC und Joyce/PCW vernachlässigen werden, denn: Bei uns kommt jeder auf seine Kosten!

In diesem Sinne und natürlich in der Hoffnung, daß Sie sich weder eine schwarze Katze noch einen Virus einfangen, verbleiben wir bis zum nächsten Monat und wünschen Ihnen nun viel Freude bei der Lektüre der neuen PC Amstrad.

Ihr

Joachim Freiburg

Joachim Freiburg

INHALT

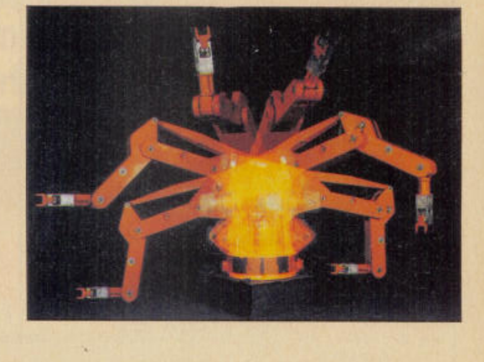
BERICHTE:

Das moderne Babel: Die Masse der Programmiersprachen ist erdrückend

S. 8

KURS:

PROGRAMME:



Mit dem Selbstbau-Interface ist auch die Steuerung eines Roboters möglich

S. 28

HARDWARE:

ASSEMBLER:

TIPS & TRICKS:



Auf diese Steine können Sie bauen

S. 16

ABENTEUER:

Januar '90



Geben Sie die Steuerzeichen direkt über die Tastatur ein

S. 63



Auch ein Schutz vor Viren: Grundlage zur effektiveren Vorsichtsmaßnahmen zeigen wir Ihnen in unserem Artikel

S. 83



In Farben und Formen schwelgen könne Sie mit Deluxe Paint II enhanced

S. 88

PCW/JOYCE

PC:

RUBRIKEN:

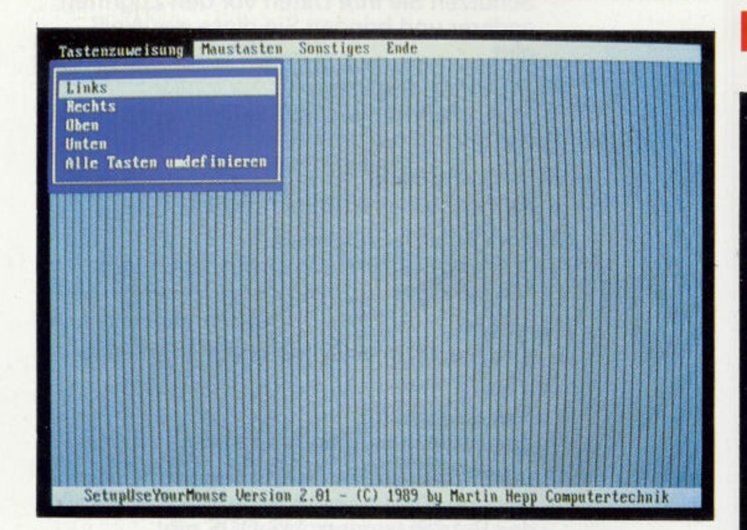
Wir waren dabei



Um den Wissensdurst der herüberströmenden Menschenmengen aus den DDR in Sachen Computer stillen zu können, stellten sich viele Mitglieder des Daten- und Medien-Verlages vom 16. bis zum 19.11.1989 mit einem Stand in die Innenstadt Eschweges und verkauften Zeitschriften, welche im Daten- und Medien-Verlag erschienen sind.

Für drei Mark Ost konnte eine Zeitschrift beziehungsweise für 10 Mark Ost konnten fünf verschiedene Zeitschriften aus dem Hause DMV käuflich erworben werden. Insgesamt wurden 12000 Mark Ost und ein kleiner Betrag in DM eingenommen. Das Geld wird den kirchlichen Kindergärten in der DDR zur Verfügung gestellt.

Use Your Mouse



Ob Sie Ihre Maus benützen oder nicht, sei Ihnen überlassen. Auf jeden Fall gibt es jetzt ein weiteres Programm, welches es Ihnen ermöglicht, Programme, die eigentlich nicht für die Mausbedienung geschrieben wurden, mit der Maus zu steuern. Dies wird durch eine Umlenkung bestimmter Tasten auf entsprechende Mausbewegungen beziehungsweise Mausklicks bewerkstelligt. Damit das Programm auch weiß, welche Tasten auf die Maus umgelenkt werden sollen, können Sie dies über das mitgelieferte Setup-Programm einstellen. Haben Sie alle Eingaben korrekt gemacht, wird ein lauffähiges Programm abgespeichert, welches nach dem Aufruf oben genanntes bewirkt. So können Sie in jedes Programm die Maussteuerung einbauen.

Info:
Martin Hepp Computertechnik,
Adolf-Guckes-Weg 1,
6239 Eppstein.

Neue Hotline-Nummern bei Microsoft

Sollte Ihnen Ihr Fachhändler bei Problemen mit Microsoft-Produkten nicht weiterhelfen können, haben Sie die Möglichkeit, von Montag bis Freitag zwischen 8.40 und 12.30 Uhr sowie zwischen 13.00 und 17.00 Uhr direkt bei der Hotline von Microsoft anzurufen.

Hier die für Sie interessanten Nummern, die nach Produkten geordnet sind:

089-31705-82: Multiplan, Excel, Chart, Works
089-31705-83: Word und Druckertreiber, Write für Windows
089-31705-84: Flugsimulator, Windows Utilities, Paintbrush
089-31705-85: Windows, Project, Maus, Netzwerk-Applikationen, Extended/Expanded Memory und LIM
089-31705-86: Compiler, Quick Sprachen

Schulversion von TIMEWORKS

Auch Schulen wurden bisher von viel zu hohen Preisen der Programme abgeschreckt, wenn es um das Thema "Desktop Publishing" ging.

Da auch diese Anwendung in Schulen beschrieben und erlernt werden sollte, bietet Schneider Data jetzt ein Vierer-TIMEWORKS-Paket für Schulen an, das zu einem Preis von 996,– DM erstanden werden kann. Pro Arbeitsplatz belaufen sich also die Kosten nur noch auf 249,– DM.

Neben dem eigentlichen Programm werden noch die leistungsstarke Benutzeroberfläche "GEM" sowie zahlreiche Druckeranpassungen mitgeliefert. Vorausgesetzt wird ein IBM-kompatibler Computer mit mindestens 512 kByte Speicher sowie zwei 360-kByte-Laufwerken oder einem 720-kByte-Laufwerk.

Weitere Informationen bei:
Schneider Data,
Postfach 1341,
8050 Freising.

Optimaler Virusschutz?

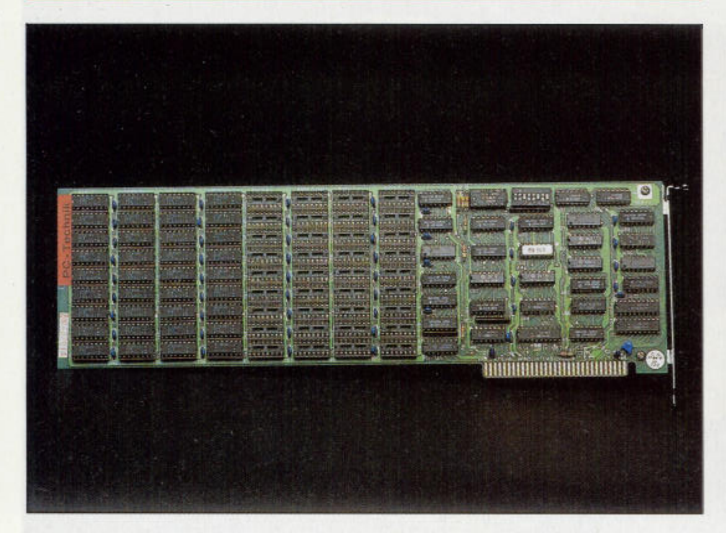


Unter dem Namen "AVSearch" wird jetzt ein weiteres Antivirusprogramm vertrieben, das in der derzeitigen Version in der Lage ist, 41 verschiedene Viren zu erkennen. Dieser Schutz gilt auch dem Israel- sowie dem Datacrime-Virus, die erst Ende letzten Jahres aufgetaucht sind.

Das Arbeiten mit dem Programm erwies sich als sehr leicht und komfortabel. So überprüft das Programm bei einfachem Aufruf ohne Parameterangabe alle Dateien auf dem aktuellen Laufwerk. Durch Parameter kann jedoch festgelegt werden, auf welchem Laufwerk, in welchem Unterverzeichnis welche Datei(en) überprüft werden sollen.

Genaueres hierzu erfahren Sie bei H+B EDV, Olgastraße 4, 7992 Tettnang 1

EMS-Speicher auch für PC1512/1640



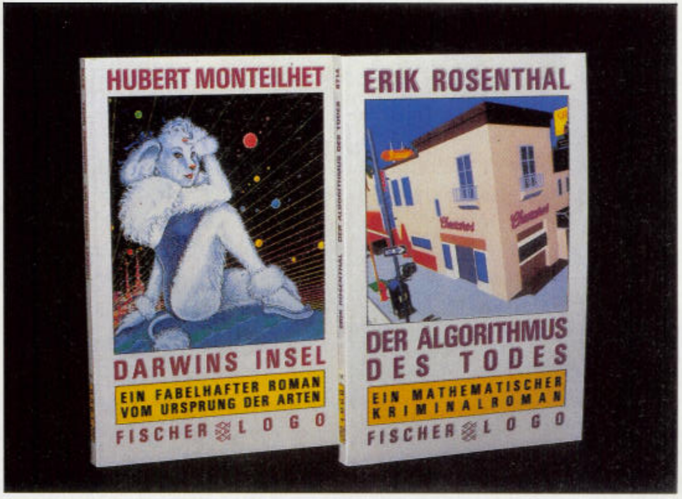
Wem seine 512 beziehungsweise 640 kByte Hauptspeicher zu wenig sind, oder wer ganz gerne einen Cache-Speicher hätte, jedoch nichts von seinem bisherigen Speicher einbüßen möchte, sollte sich überlegen, ob für ihn die EMS-Karte nicht eine lohnenswerte Anschaffung ist. Insgesamt können vier dieser mit jeweils 2 MByte bestückten Karten installiert werden, so daß Sie bei dieser Konfiguration ganze 8 MByte als Extended oder Expanded Memory nutzen können.

Die Installation der Karte ist völlig problemlos, da die Karte nur eingesteckt werden muß und später durch mitgelieferte Programme installiert werden kann.

Im Lieferumfang ist außer einem deutschen Handbuch und der Betriebssoftware auch noch ein gutes Festplattenbackup-Programm enthalten.

Info:
PC-Technik,
Rheinstraße 16,
5202 Hennef 1

Büchertip



Wer gern um die Ecke denkt, eine Vorliebe für Tüfteleien hat oder einfach einmal einen guten Roman lesen möchte, ist sicherlich mit 'Der Algorithmus des Todes' oder 'Darwins Insel' bestens bedient. In diesen Romanen ist all das vereint, was einem Computerbesitzer und Tüftler das Herz höher schlagen läßt.

Erschienen sind die Bücher im Fischer Taschenbuch Verlag. Sie sind in jeder guten Buchhandlung erhältlich.

Public Domain

Die Firma Boeder, die bisher hauptsächlich für Ihre Disketten bekannt war, bietet diese jetzt auch in bespielter Form an. Zu einem Preis von 9,95 DM bis 19,98 DM können bis jetzt aus einer Palette von rund 100 Programmen aus der Public Domain beziehungsweise Shareware-Programme ausgewählt werden. Nach Auskunft von Boeder soll diese Palette jedoch bis 1990 auf 200 Programme für PC, C64, AMIGA und ATARI aufgestockt werden.

Info:
Boeder,
6093 Flörsheim am Main

Softfonts verwalten

Star Division bietet ab sofort das Programm Laser Star 1.0 an, welches für die Verwaltung von Download-Softfonts und Fontcartridges druckerinterne Schriften übernimmt. Es stellt die gewählten Schriften im Drucker ein und erstellt automatisch Druckertreiber für MS-WORD, WordPerfect, MS-Windows und Star Writer 5.0. Mit Hilfe des mitgelieferten Font-Generators ist es ebenfalls möglich, jeden SoftFont beliebig zu verändern.

Abgerundet wird das Bild durch 50 mitgelieferte Softfonts aus verschiedenen Schrift-Familien.

Info:
Star-Division,
Zum Elfenbruch 5-11,
2120 Lüneburg.

Laufwerke jetzt billiger

Bei den Diskettenlaufwerken für den CPC der Firma Frank Strauß Elektronik gibt es jetzt einige Veränderungen.

Ab sofort verfügen alle Laufwerke über einen Seitenumschalter, damit dem Benutzer unter AMSDOS sowie unter CP/M 2 x 360 kByte Speicherkapazität zur Verfügung stehen. Die Preise für diese Laufwerke wurden für das 5,25-Zoll-Format auf 323,– DM und für das 3,5-Zoll-Format auf 279,– DM reduziert.

Info: Frank Strauß Elektronik,
Schmiedstraße 11,
6750 Kaiserslautern

Lernen mit dem Computer

Wer sich einen Computer kauft, will sicherlich nicht nur mit ihm spielen, sondern wurde sicherlich von dem Hinter- beziehungsweise Hauptgedanken getragen, sich an diesem weiterzubilden.

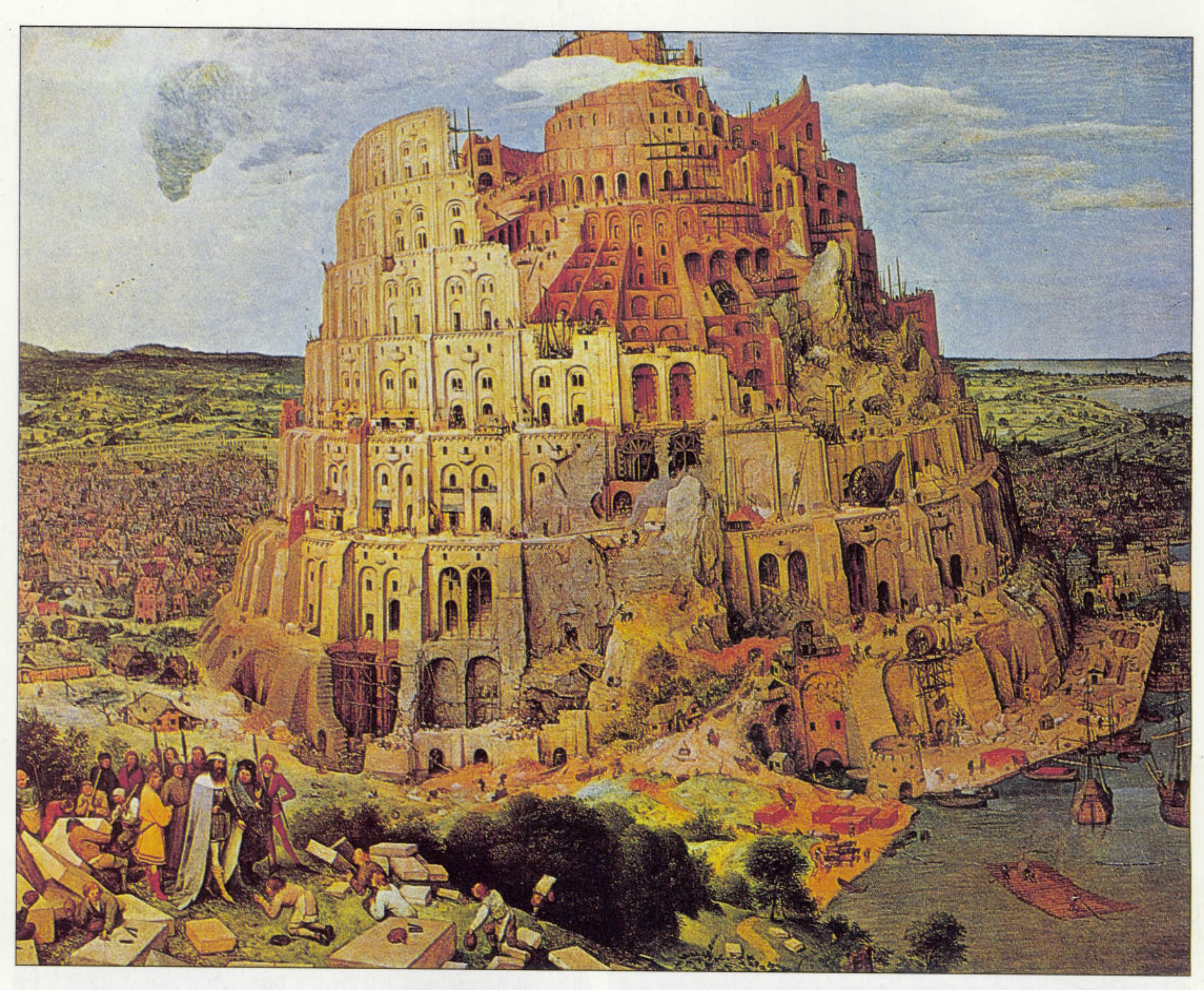
Das diese Bildung nicht nur stur in Richtung Vokabeltrainer geleitet werden kann, wird bei dem Arbeiten mit den Lernprogrammen von Kienitz + Grabis klar. Für einen erstaunlich niedrigen Preis von 10,– DM wird jeden Monat ein weiteres Programm erscheinen. Bisher sind folgende Themen lieferbar:
Mit dem PC auf Du und Du, Englisch auf Reisen, Auto und Verkehr.
Da diese Kurse nur in limitierter Auflage erscheinen werden, sollten Sie sich beeilen.
Nähere Informationen hierzu können Sie bei folgender Adresse einholen:
Kienitz + Grabis - Soft- + Hardware-Versand + Verlag,
Schulstraße 18,
8913 Schondorf.

Flexible Teak-Laufwerke

Die Diskettenlaufwerke vom Typ Teak für CPC und PCW der Firma Frank Strauß Elektronik lassen sich bei einem Umstieg auf XT- bzw. AT-kompatible Computer in der Regel weiternutzen. Meist reicht eine Veränderung der Jumpereinstellung aus, um gewünschten Nutzen zu erreichen.

Auskunft bei der Umrüstung ist zu erhalten bei:

FSE Frank Strauß Elektronik
Schmiedestraße 11
6750 Kaiserslautern



Babylonisches Sprachgewirr

In welcher Sprache soll ich programmieren? Programmiere ich in der falschen Sprache? Was ist die Sprache der Zukunft? Solche und ähnliche Fragen stellen sich allen Computerbesitzern, die ihre Programme selbst entwickeln wollen. Hier soll nun versucht werden, einen kleinen, beileibe nicht vollständigen Überblick über bekannte, und manche vielleicht unbekannte Programmiersprachen zu geben.

In der mittlerweile etwa dreißigjährigen Geschichte der elektronischen Datenverarbeitung gibt es eine Menge guter und wohldurchdachter Sprachkonzepte. Sehr grob könnte man diese Konzepte wie folgt gliedern:
- Maschinensprache
- höhere Programmiersprachen
- objektorientierte Sprachen
- KI-Sprachen
- spezielle Programmiersprachen.

In dieser Reihenfolge werden nun auch die verschiedenen Programmiersprachen vorgestellt.
Die einzige Sprache, die der Prozessor Ihres Computers versteht, sieht so aus:
0101011100011111

Unverständlich? Im Vertrauen, ich habe das auch nur aus einem Assemblerbuch abgeschrieben.

Die Maschine und ihre Sprache

Vor dem gleichen Problem (was soll das denn schon wieder bedeuten?) standen auch in der Vergangenheit die Programmierer. Damals mußten die Programme noch mühevoll in Binär-, Oktal- oder Hexadezimalzahlen aufgeschrieben werden, um dann mit Hilfe eines Ladeprogramms in den Rechner gebracht zu werden. Natürlich hat dieses Verfahren, Programme zu entwickeln, sehr viele Fehlermöglichkeiten. Deshalb wurde auch recht schnell die Assemblersprache entwickelt.

Assembler

Beim Assembler werden den einzelnen Befehlscodes bestimmte Namen zugeordnet, die Mnemoniks. Damit wird erreicht, daß die maschinensprachlichen Programme lesbarer und dadurch auch weniger fehlerträchtig werden. Es ist für Menschen auch viel einfacher, sich kurze Befehlswörter wie

Beispiel 1: Assembler

```
code Segment
        assume cs:code, ds:code
        org 0100H
anfang:
        jmp short start
text db 'Assembler', 10, 13, '$'
start:
        mov cx, 10
        lea dx, text
        mov ah, 09H
ausgabe:
        int 021H
        loop ausgabe
        mov ax, 04c00H
        int 021H
code Ends
        end anfang
```

mov, push, usw. zu merken als binäre oder hexadezimale Zahlen. Nachteilig ist es, daß die Assemblersprache (leider wird ebenso wie die Sprache selbst auch der Übersetzer, der letztendlich wieder reine Maschinensprache liefert, Assembler genannt) nur auf einen bestimmten Prozessor zugeschnitten ist. Wer also auf seinem alten Z80-System dessen Assemblersprache gelernt hat, kann beim Umstieg auf einen PC alles wieder vergessen und neu mit dem Lernen beginnen.

Wer die Maschinensprache der 80x86-Prozessoren jedoch beherrscht, steht praktisch mit dem PC auf Du und Du. Keine andere Sprache besitzt so viele Freiheiten und Gefahren. In keiner anderen Sprache lassen sich gleich schnelle Programme entwickeln. In keiner anderen Sprache dauert die Programmentwicklung so lange und sind so viele Rechnerabstürze an der Tagesordnung wie in Assembler. Dennoch führt an Assembler kein Weg vorbei, wenn es um ganz bestimmte Aufgaben geht.

Gerätetreiber für das DOS, zeitkritische und in Echtzeit reagierende Routinen lassen sich praktisch nur in Assembler schreiben. Komplette Programme werden jedoch nur noch selten in dieser Sprache programmiert, bei bestimmten Funktionen bietet sich Assembler jedoch an. Man kann also sagen, daß die Assemblersprache zur Ergänzung einer Hochsprache ideal ist.

Höhere Programmiersprachen

Damit sind wir also bei den höheren Programmiersprachen angekommen. Praktisch ist alles, was nicht wie Assembler mit den Bits, Bytes und Registern im Rechner direkt herumspielt, eine höhere Programmiersprache. Man trennt jedoch diese Sprache in verschiedene Untergruppen auf. Fangen wir hier also mit den 'normalen höheren' Programmiersprachen an.

FORTRAN

W. J. Backus entwarf in den Jahren 1954 bis 1957 den Dinosaurier unter den Programmiersprachen, FORTRAN (FORmula TRANslation), die damit die älteste (zusammen mit LISP) aller höheren Sprachen ist. FORTRAN wurde entwickelt, um besonders einfach numerische Probleme mit zahlreichen Formeln und Variablen darstellen zu können. Diese Ausrichtung auf numerische Probleme läßt sich auch an der Einfachheit des Umgangs mit reellen und komplexen Zahlen sowie Vektoren und Matrizen erkennen. Viele einfache mathematische Funktionen sind bereits in die Sprache integriert (Expotentialfunktionen, Logarithmus, usw.). Ein Programm in FORTRAN besteht aus einem Hauptprogramm und Subroutinen. Variablen müssen nicht, wie in C oder Pascal, vor ihrer ersten Benutzung deklariert werden. Bei der Lösung von nichtnumerischen Problemen tut sich FORTRAN jedoch schwer. Vielfach wird an FORTRAN auch das Fehlen von Mitteln zur strukturierten Programmierung kritisiert.

FORTRAN ist dessen ungeachtet jedoch im wissenschaftlichen Bereich auf Großrechnern immer noch die Nummer eins. Das erklärt sich vor allem aus den reichhaltigen Programmbibliotheken für Probleme im naturwissenschaftlichen Bereich, die es in FORTRAN gibt. Auch wird derzeit an einer Revision FORTRANs gearbeitet (FORTRAN 8x, was wohl doch eher ein FORTRAN 9x wird), die spezielle Möglichkeiten zur Unterstützung der Parallelprogrammierung auf Multiprozessoranlagen besitzen soll.

Es gibt auch einige PC-Implementationen von FORTRAN, doch durch die wissenschaftliche Ausrichtung ist die Verbreitung im PC-Sektor recht gering.

COBOL

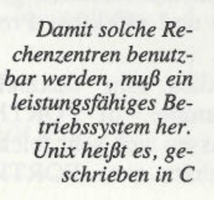
Während einer Konferenz 1960 einigten sich Vertreter der amerikanischen Regierung und der Industrie auf einen Standard für die Sprache COBOL (was für COmmon Business Oriented Language steht). Mit diesem Namen ist eigentlich auch schon gesagt, wo diese Sprache hauptsächlich eingesetzt wird: im Geschäftsbereich. COBOL verfügt deshalb über eine Dezimalarithmetik, stellt viele Datenstrukturen zur Verfügung und kann mit großen Datenmengen und Dateien problemlos umgehen. Die COBOL-Syntax ist sehr ausführlich und liest sich fast wie Klartext, was diese Sprache aber auch zu einem Schwätzer unter den Programmiersprachen macht. Etwas wie **a = c / b** wird in Cobol als **divide c by b giving a** geschrieben. Das hat aber auch den Vorteil, daß COBOL-Programme fast selbstdokumentierend sind.

COBOL ist mit eine der am meisten verwendeten Sprachen auf Großrechnern. Es gibt zwar verschiedene Implementationen für Großrechner und PCs, der Sprachumfang aber ist durch eine strenge Normung bei allen fast gleich. COBOL hat auch keine hardwarespezifischen Elemente. In Betrieben, die Großrechner und PCs einsetzen, kann der Schulungsaufwand deshalb recht gering gehalten werden.

ALGOL

ALGOL (ALGOrithmic Language) ist ein europäisches Werk. Entwickelt wurde sie unter anderem von C.A.R. Hoare (Erfinder des Quicksort-Algorithmus) und N. Wirth, der später auch Sprachen wie PASCAL und MODULA II entwickelte. ALGOL-60 wurde im Jahre 1960 entwickelt. Eine erweiterte Version, ALGOL-68, entstand. Wie man bei dem Namen vermuten kann, etwa 1968. ALGOL ist blockorientiert und besitzt schon alle wichtigen Kontrollstrukturen (IF-THEN-ELSE, CASE, REPEAT, WHILE), die zu einer modernen Programmiersprache gehören. ALGOL kennt auch Unterprogramme, die als Prozeduren oder Funktionen realisiert werden. Diese Unterprogramme können lokale Variablen haben, so daß die Speicherverwaltung nicht mehr rein statisch ist. Die





Damit solche Rechenzentren benutzbar werden, muß ein leistungsfähiges Betriebssystem her. Unix heißt es, geschrieben in C



Blaise Pascal, der Namensgeber der wohl bekanntesten Programmiersprache, wird wohl wenig von seinem späteren Ruhm geahnt haben

formatgebundene Syntax, wie bei FORTRAN ein Relikt aus der Lochkartenära, fällt bei ALGOL weg. Viele moderne Sprachen sind Abkömmlinge von ALGOL, die als erste Programmiersprache gelten kann, die nach informationstheoretischen Gesichtspunkten entwickelt wurde.

APL

APL, was einfach A Programming Language heißt, gehört, trotz leistungsfähiger Implementationen, innerhalb des PC-Marktes noch zu den Exoten. Im industriellen Bereich kann sich APL jedoch in der Verbreitung mit COBOL messen. In den 60er Jahren wurde APL von K. Iverson an der Harvard Universität entworfen. Es handelt sich bei APL um eine speziell zur Darstellung mathematischer Probleme entworfene Sprache.

Ursprünglich von Iverson als Notationssystem für Mathematikvorlesungen gedacht, enthält APL eine Unzahl aus der Mathematik stammender 'Hieroglyphen'. Das, und der weitestgehende Verzicht auf Schleifen, machen das Einarbeiten in APL sehr mühsam und zeitintensiv. Dennoch ist diese Sprache nicht nur für reine Mathematiker interessant. APL besitzt spezielle Operatoren für Vektor- und Matrizenrechnungen, die dadurch sehr effizient ablaufen. Beim Umgang mit mathematischen Strukturen ist APL, was Formulierung und Performance angeht, so ziemlich ungeschlagen.

FORTH

Wie APL ist FORTH ein Kind der sechziger Jahre. Entwickelt wurde es von C.H. Moore zur Maschinensteuerung und für Echzeitanwendungen. FORTH stellt eine Art Maschinensprache dar, die jedoch nicht auf einen bestimmten, sondern auf einen virtuellen (simulierten) Prozessor läuft. Inzwischen gibt es aber auch FORTH-Prozessoren, die FORTH als ihre Maschinensprache besitzen.

In einer solchen Umgebung ist FORTH natürlich sehr schnell; auf einem normalen Rechner ist FORTH allerdings auch nicht gerade langsam. FORTH ist eine stackorientierte Sprache. Sie besitzt zwei Stacks, den normalen Returnstack, der die Rückkehr-Adresse während eines Unterprogrammaufrufs (FORTH: eines Wortes) aufnimmt, und einen Daten-Stack, der für sämtliche Operationen mit Daten zuständig ist.

Es wird in FORTH mit der sogenannten umgekehrt polnischen Notation gearbeitet (wie bei einigen wissenschaftlich-technischen Taschenrechnern). Dies läßt sich an einem kurzen Beispiel wohl am besten erklären.

Was in den meisten Programmiersprachen als

a = 4 + 5

geschrieben wird, sieht in FORTH so aus:

4 5 +

Das bedeutet folgendes: zuerst wird 4 auf den Datenstack abgelegt, danach die 5. Dann werden die beiden obersten Werte des Datenstacks addiert (+), der Stackpointer entsprechend verändert, und das Ergebnis als oberster Wert wieder auf dem Datenstack abgelegt.

Da es zahlreiche Möglichkeiten in FORTH gibt, den Datenstack zu manipulieren, lassen sich, wenn man sich an die umgekehrt polnische Notation gewöhnt, sehr kurze und effektive Programme schreiben.

FORTH ist überdies sehr flexibel. Neue Unterprogramme, in FORTH Wörter genannt, lassen sich sehr leicht dem Grundwortschatz von FORTH hinzufügen. Nachteil ist natürlich die dadurch bedingte Neigung zu Dialekten, doch ist für FORTH in nächster Zukunft ein ANSI-Standard angekündigt. FORTH ist leicht auf neue Rechner zu implementieren, da der High-Level-Code auf relativ wenigen Maschinenroutinen basiert.

BASIC

BASIC, was für Beginners All purpose Symbolic Instruction Code steht, ist wohl die am meisten verbreitete und verwendete Sprache auf Klein-Computern. Bei fast jedem verkauften Computer, ob PC, CPC, C64 usw., gehört eine BASIC-Version zum Verkaufsumfang. Bei PCs ist es ein gesondert zu ladender Interpreter (etwa GW-BASIC), bei manchen Home-Computern (CPC) ist es zugleich im Betriebssystem integriert.

BASIC wurde 1960 von J. Kemmeny und Th. Kurtz als FORTRAN-ähnliche Interpretersprache konzipiert. Ein Vorteil von BASIC ist die leichte Erlernbarkeit. Vorteilhaft ist auch, daß diese Programmiersprache meistens im Preis des Rechners enthalten ist. Allerdings erlaubt BASIC in den meisten Versionen keine strukturierte Programmierung, und es fehlen lokale Variablen und echte Unterprogramme. Ebenso fehlt bei den meisten Dialekten die Möglichkeit, eigene Datenstrukturen anzulegen.

Beispiel 2: Basic

```
10 FOR I=1 TO 10 STEP 1
20 PRINT "Basic"
30 NEXT I
```

In letzter Zeit sind jedoch verstärkt Dialekte (Turbo BASIC, Quick BASIC) aufgetreten die mit diesen Mängeln aufräumen. Sie sind dabei doch so Pascal–ähnlich geworden, daß man sich fragt, warum man nicht gleich Pascal benutzt.

Pascal

Pascal ist die wohl bekannteste Sprachentwicklung von Niklaus Wirth. Sie wurde von ihm 1970 als Lehrsprache für den Informatikunterricht entwickelt. Durch ihre Leistungsfähigkeit setzte sie sich aber auch in der professionellen Programmentwicklung durch. Pascal enthält alles, was eine gute Programmiersprache auszeichnet. Es lassen sich neue Datentypen definieren, es gibt Unterprogramme in der Form von Funktionen und Prozeduren, lokale

Variablen, die Möglichkeit Unterprogramme rekursiv, also sich selber aufrufend, zu gestalten und vieles mehr. Pascal enthält auch ein Zeigerkonzept zur dynamischen Erzeugung von Daten und Datenstrukturen während der Laufzeit eines Programms.

Den Siegeszug auf kleinen und mittleren Rechnern trat Pascal an, als K. Bowles UCSD-Pascal entwickelte. Obwohl es den 'Wirth-Jensen-Standard' und den ISO-Standard (ISO = International Standards Organisation) für Pascal gibt, entstanden zahlreiche Dialekte. Ein De-facto-Standard stellt Turbo Pascal von Borland dar, das 1983 für MS-DOS- und CP/M-Systeme erschien, und innerhalb eines Jahres 300000mal verkauft wurde. Besonders gefördert wird von Pascal ein klar strukturierter Programmierstil. Im Gegensatz zu BASIC muß der Programmierer sich erst Gedanken über das Problem und seine Lösung machen, anstatt ins Blaue hinein zu programmieren.

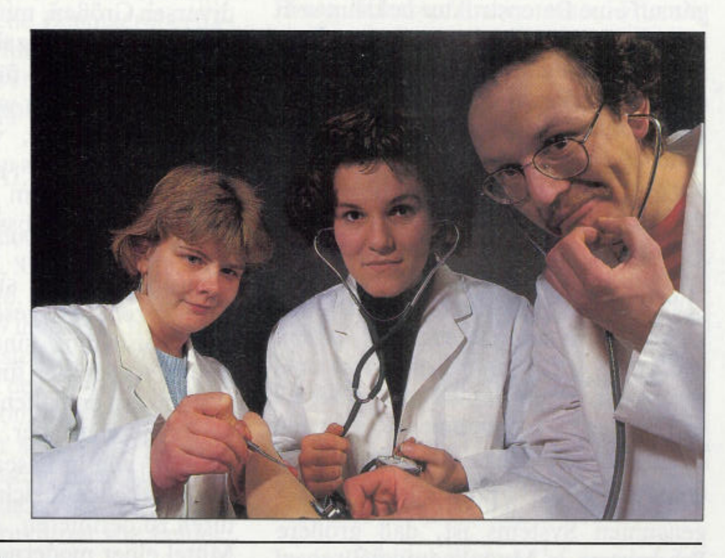
Beispiel 3: Pascal

```
Program Beispiel;
var
   i : integer;
begin
   for i:=1 to 10 do
   writeln( 'Pascal' );
end.
```

Modula II

Modula II wurde, wie Pascal, von N. Wirth an der ETH Zürich entwickelt. Wirth definierte etwa 1975 eine Weiterentwicklung von Pascal: Modula, die zum Ende der siebziger Jahre, im Laufe eines Forschungsprojekts, zu Modula II ausgearbeitet wurde.

In diese Sprache flossen zahlreiche neue Software-technologische Konzepte, wie Modularisierung von Programmprojekten und die Trennung von 'öffentlichen' und 'privaten' Daten und Algorithmen, ein. Modula II enthält bereits in der Sprachkonzeption die Möglichkeit, 'Co-Prozesse' (parallele oder quasiparallele Ausführung von Unterprogrammen) zu formulieren.



Die Arztvisite: Machen Expertensysteme, geschrieben in Prolog, solchen Dienst am Patienten in der Zukunft überflüssig?

Ein Modul (Programmbaustein) in Modula II, daher der Name, besteht aus zwei Teilen: dem Definitionsteil und dem Implementierungsteil.
Beide Teile können unabhängig voneinander übersetzt werden (mit gewissen Einschränkungen). Im Definitionsteil ist genau aufgeführt, welche Variablen, Definitionen, Funktionen und Prozeduren des Moduls nach außen hin bekannt sind. Das geht soweit, daß innerhalb eines Programms nur die Zeiger auf eine Datenstruktur bekannt sein müssen, nicht mehr die Struktur selbst. Diese wird innerhalb des Moduls vereinbart, und kann auch nur dort mit den im Modul enthaltenen Funktionen verändert werden.

Beispiel 4: Modula

```
MODULE Beispiel;
FROM InOut IMPORT WriteLn, WriteString;
VAR I : CARDINAL;
BEGIN
  FOR I := 1 TO 10 DO
  BEGIN
    WriteString("Modula");
    WriteLn;
  END;
END Beispiel.
```

Vorteile dieses 'Informationhidings' genannten Systems ist, daß größere Programme in Module aufgeteilt werden können. Sind mehrere Programmierer an solch einem Projekt tätig, reicht es aus, daß nur der Definitionsteil der Module bekannt sein muß. Der Compiler prüft beim Zusammenbau eines Programms anhand der Daten im Definitionsteil genau die Typenverträglichkeit bei Parametern von Funktionen und der richtigen Benutzungen von Variablen.
Zusammenfassend gesagt: Modula II ist eine Universalsprache, die ihre besonderen Stärken bei größeren Projekten hat. Die Bibel aller Modula-Programmierer ist das Buch 'Programming in Modula-II' von N. Wirth.

C

C wurde etwa 1972 von Dennis M. Ritchie entwickelt. C ist eine Weiterentwicklung der Sprache B, die ihrerseits aus BCPL entstand. C hat eine Menge Sprachelemente, mit diesen Sprachen gemeinsam ist jedoch kein Dialekt. B und BCPL sind typenlose Sprachen, deren einziger Datentyp das Maschinenwort ist. In C gibt es jedoch viele vordefinierte Typen (Integer-Zahlen in diversen Größen, mit und ohne Vorzeichen, Gleitkommazahlen etc.).
C ist eine Sprache für allgemeine Probleme. Anfänglich war C sehr eng mit UNIX verbunden, was nicht zuletzt daran liegt, daß das ursprünglich in Assembler auf einem PDP-11-Rechner entwickelte Betriebssystem später in C übertragen wurde. UNIX und seine Dienstprogramme sind heute fast zu 100 Prozent in C geschrieben. Die einzige Ausnahme sind einige hundert Zeilen Assembler für die im Betriebssystemkern enthaltenen sehr hardwarenahen Gerätetreiber.
Es gibt, wie in Pascal und Modula II, die Möglichkeit, sich neue Datenstrukturen zu definieren. Auch alle anderen Mittel einer modernen Sprache sind in C enthalten, z.B. Zeiger, Adreßarithmetik, rekursiver Funktionsaufruf, lokale Variablen, diverse Kontrollstrukturen usw.
C ist, im Gegensatz zu den meisten anderen Programmiersprachen, nicht von Gremien, sondern von Programmierern entwickelt. Merkwürdigerweise (oder auch nicht) sind C-Compiler untereinander auch auf verschiedenen Rechnersystemen erheblich kompatibler als es bei 'genormten' Sprachen der Fall ist.
Während C-Compiler früher dem Programmierer alle Freiheiten ließen (nach dem Motto: die/der Frau/Mann wird schon wissen, was sie/er tut), sind neuere Compiler wesentlich rigoroser. Die sogenannten ANSI-Compiler (nach dem ANSI-C-Standard -ANSI = American National Standards Institute, etwa wie DIN) sind sehr streng in der Überprüfung der Konsistenz von Parameterübergaben und ähnlichem.

Beispiel 5: C

```
#include <stdio.h>
void main( void )
{
    int i;
    for( i = 0; i < 10; i++ )
    printf( "C\n" );
}
```

Die oft geäußerte Meinung, C wäre sehr schwer zu erlernen, ist schlichtweg falsch. C hat nur einen sehr kleinen Sprachumfang (etwa 30 reservierte Worte), was die Erlernbarkeit sehr erleichtert. Die schwere Lesbarkeit, die man C-Programmen nachsagt, liegt meist an den knappen Formulierungen dieser Sprache und daran, daß die meisten C-Freaks einen kryptischen Programmstil kultivieren (nach dem Motto: Wir sind was Besonderes).
Moderne C-Compiler erzeugen Programme, deren Größe und Laufzeitverhalten kaum hinter guten Assemblerprogrammen zurückstehen. Zusammenfassend kann man sagen, daß C die Sprache ist, die die Geschwindigkeit und Fähigkeiten von Assembler mit den Vorzügen einer Hochsprache zusammenbringt.

ADA

Etwa um 1973 begann das DoD (Department of Defense = Amerikanisches Verteidigungsministerium) Studien, mit dem Ziel, eine einheitliche NATO-Programmiersprache zu erstellen. 1975 wurde dann mit der Entwicklung dieser Sprache begonnen. 1977 waren die Entwürfe von vier Sprachen verschiedener Firmen fertig. Sie wurden von Vertretern verschiedener Institutionen geprüft und bewertet. Der siegreiche Entwurf wurde dann ADA (zu Ehren von Lady Ada Augusta, Countess of Lovelace) genannt.
Nach weiteren drei Jahren war die vorläufige ADA-Definition fertig. Es dauerte aber nochmal drei Jahre, bis die endgültige Fassung des 'ADA Language Reference Manual' (ALRM) fertig war. Seit 1986 ist ADA die offizielle Programmiersprache der NATO und das DoD wacht über die Einhaltung des Sprachstandards (nur ein vom DoD überprüfter Compiler darf sich ADA-Compiler nennen).

BASIC auf dem CPC. Viele Vor- und Nachteile machen diese Sprache zum Streitobjekt unter manchen Programmierern

Soweit die Vorgeschichte dieser Sprache. Doch was kann diese Sprache, die alle anderen ablösen soll? ADA ist zum Beispiel echtzeitfähig. Das Modulkonzept von ADA geht weit über das von Modula II hinaus. Die Entwicklung von selbstdokumentierenden Programmen wird ebenso unterstützt, wie die hardwarenahe Programmierung durch Einfügen von Maschinenbefehlen. Ein ADA-Compiler nimmt sehr genaue Prüfungen der Typkonsistenz vor. Er ist hier noch pingeliger als ein Modula-Compiler.

Dies, und noch mehr, spricht für ADA, dagegen spricht der derzeit sehr hohe Preis (im Vergleich zu anderen Compilern wie C und Pascal) und die Mächtigkeit der Sprache. Warum die Mächtigkeit, werden sie fragen. Ganz einfach: Ein guter C-Compiler besteht aus etwa 20.000 Instruktionen, ein ADA-Compiler aus 200.000.

Objektorientierte Sprachen

Einfach gesagt handelt es sich bei Objekten um eine Menge von Daten und Prozeduren, die diese Daten beeinflussen. Eine nicht zum Objekt gehörende Prozedur kann nicht auf den Objektinhalt zugreifen. Diese Kapselung von Daten hat viele Vorteile. Die objektinternen Daten sind zum Beispiel vor Seiteneffekten und Mißbrauch geschützt.

Es ließe sich noch wesentlich mehr über die Grundlagen dieser Sprachen sagen, doch reicht der Platz hier nicht aus, so daß auf weiterführende Literatur verwiesen werden muß.

Übrigens werden Sie wohl schon, ohne es zu bemerken, mit der objektorientierten Programmierung zusammengetroffen sein. Sierra, zum Beispiel, die Entwickler des Spiels Leisure Suit Larry, benutzen objektorientierte Entwicklungswerkzeuge für ihre Adventures.

SMALLTALK

SMALLTALK ist wohl der wichtigste Vertreter der objektorientierten Sprachen. SMALLTALK entstand in den siebziger Jahren im XEROX-PARC (Palo Alto Research Center). Es war der Versuch, Betriebssystem, Sprache, Editor und grafische Benutzeroberfläche unter einen Hut zu bringen. Aus dieser Benutzeroberfläche entwickelten sich später Apple's Finder, Gem, Windows usw.

Die meisten SMALLTALK-Entwicklungsumgebungen sind Interpreterähnlich, haben jedoch ein recht gutes Laufzeitverhalten. SMALLTALK bietet eine aufwendige, interaktive Benutzerschnittstelle, Windows, Grafikeditor, Icons, Maussteuerung, die, wie alle anderen Teile des Systems, in Anwendungen übernommen werden können. Es gibt auch sehr viele vordefinierte Datenstrukturen, mehr als bei allen anderen Sprachen. Da wären z.B. die klassischen Strukturen wie Felder, Records, Listen, Mengen ... und auch Ungewöhnlicheres wie Assoziationen, Symbole, Dictionaries und noch vieles mehr.

Ansonsten ist zu SMALLTALK noch zu bemerken, daß es sich hierbei, wie auch bei C oder Modula II, um eine Sprache für allgemeine Anwendungen handelt. Anzumerken sei auch noch, daß, wer mit dem Gedanken spielt, sich ein SMALLTALK-System zuzulegen, über einen schnellen Rechner mit viel Speicher und einer großen Festplatte verfügen sollte.

C++

Es handelt sich bei C++ praktisch um C, das um Objekte erweitert wurde. Entwickelt hat dies Bjarne Stroustrup im Jahre 1985. Die Verbund-Datenstruktur 'struct' wurde vom ihm zu der Struktur 'class' erweitert. Diese 'class' kann neben Datenelementen auch Funktionsdeklarationen, Operator-Überladungen und ähnliches enthalten.

Diese 'Klassen' haben einen öffentlichen und einen versteckten Teil. Die versteckten Teile sind nur innerhalb der Klasse zugänglich. Andere Programmteile können nur auf die 'öffentlichen' Teile einer Klasse zugreifen. Sehr flexibel wird das Ganze durch die Möglichkeit, Operatoren zu überladen. Das heißt, daß es möglich ist, innerhalb einer 'Klasse' Operatoren als 'Methoden' (Funktionen) zu definieren, – hierbei kommen alle aus C bekannten Operatoren in Frage – die dann, auf Objekte ihrer Klasse angewendet, ihre normale Bedeutung verändern. So könnte man '+' zum Addieren von kompletten Matrizen verwenden, indem man dies für diese Klasse deklariert.

In der Bibliothek von C++ wird der Operator '<<' als Ausgabeoperator für Streams verwendet. Durch die mehrfache Definition dieser Operators ist es dann möglich, fast jeden beliebigen Parameter auszugeben.

C++ erfüllt die Grundvoraussetzungen einer objektorientierten Sprache, bleibt jedoch weit hinter den Möglichkeiten von SMALLTALK zurück. Für C-Programmierer, die nun auch Objekte behandeln wollen, bietet C++ aber alle Möglichkeiten für einen schnellen und effektiven Umstieg.

Pascal++

Selbst das ehrwürdige Pascal wurde von dem neuen Boom der Softwareindustrie nicht verschont und erhielt objektorientierte Erweiterungen. Mit Turbo Pascal 5.5 von Borland und Quick Pascal 1.0 (kompatibel zu Turbo Pascal 5.0) von Microsoft, traten zwei objektorientierte Pascal-Versionen auf (weitere werden wohl folgen). Da diese Erweiterungen stark an C++ angelehnt sind, gibt es eigentlich auch nicht mehr darüber zu sagen.

KI

KI-Künstliche Intelligenz-Sprachen unterscheiden sich in vielfacher Hinsicht von den normalen Programmiersprachen. Diese Sprachen behandeln Daten als Symbole, vergleichbar den Objekten der objektorientierten Sprachen. Ansammlungen von Symbolen werden zu Listen zusammengefaßt, die dynamisch verwaltet werden.
Der große Unterschied zu anderen Sprachen liegt darin, daß KI-Sprachen vornehmlich zur Lösung nichtnumerischer Probleme entwickelt wurden.

LISP

LISP (LISt Processing Language) ist, zusammen mit FORTRAN, die älteste Computer-Hochsprache. Sie wurde Mitte der fünfziger Jahre am MIT (Massachusetts Institute of Technology) entwickelt. Während FORTRAN zur Lösung numerischer Probleme erstellt wurde, steht bei LISP die Lösung nichtnumerischer Probleme im Vordergrund. Sie ist eine offene Sprache, das heißt, der Sprachumfang läßt sich einfach erweitern. Dies führte aber auch dazu, daß eine Unzahl von Dialekten entstand. 1962 setzte zwar das erste LISP-Programmierhandbuch einen Quasi-Standard, der jedoch nicht lange aufrecht blieb.
Versuche, einen Standard zu bilden, scheiterten meist daran, daß in Betracht gezogen wurde, alle Vorteile der Dialekte in den neuen Standard einzubringen. Als Beispiel hierfür kann Golden Common LISP gelten (natürlich eine Gremiensprache). Zum Sprachumfang von Golden Common LISP gehören mehrere 1000 Funktionen, so daß nur für sehr große Rechner (etwa ab 386 aufwärts) oder spezielle LISP-Maschinen Implementierungen existieren, die den vollen Sprachumfang bieten.
Einen anderen Weg ging die Quelle von LISP, das MIT. Dort wurde, etwa in der Mitte des letzten Jahrzehnts, eine LISP-Version namens Scheme entwickelt. Scheme zeichnet sich besonders durch die Rückbesinnung auf das Wesentliche und Wichtige dieser Sprache aus, so daß es gelang, Scheme kompakt und gleichzeitig sehr leistungsfähig zu machen. Dank der Kompaktheit gibt es auch leistungsfähige Scheme-Implementationen für Microcomputer.
Der Anwendungsbereich von LISP ist sehr groß, speziell aber auf nichtnumerische Problemlösungen ausgerichtet. Programme wie Expertensysteme oder intelligente Datenbankabfragen lassen sich sehr leicht in LISP realisieren.

PROLOG

Die zweite große, im KI-Bereich eingesetzte Sprache ist PROLOG (PROgramming in LOGic). PROLOG wurde 1972 in Marseille, Frankreich, erstmals implementiert. PROLOG ist eine nichtalgorithmische Sprache. Das bedeutet: In PROLOG gibt es keine Möglichkeit, den Programmfluß zu steuern. PROLOG kennt nur Funktionen, deren Wert entweder wahr oder falsch sein kann. Der Programmierer braucht nur Daten und ihre Verknüpfungen anzugeben. Das Programm ermittelt dann alle möglichen Lösungen. Es reicht also aus, ein Problem in seiner Struktur zu beschreiben, die Lösung wird von PROLOG selbst über einen eingebauten Deduktionsmechanismus gesucht.
Seinen Aufstieg verdankt PROLOG vor allen Dingen der Entscheidung japanischer Forschungsinstitute, für die nächste 'intelligente' Computergeneration als Basissprache zu dienen. Wie LISP eignet sich PROLOG speziell zur Lösung nichtnumerischer Probleme. Auch die Schwierigkeit, numerische Probleme darzustellen, haben beide Sprachen gemeinsam.

Spezielle Hochsprachen

Alle bis jetzt vorgestellten Sprachen hatten eins gemeinsam: Es sind Sprachen für jeden Verwendungszweck. Sie mögen ihre Stärken und Schwächen haben, man kann prinzipiell aber alle Probleme mit ihnen lösen. Es gibt aber auch eine Gattung der Hochsprachen, die nur auf einen bestimmten Verwendungszweck hin entwickelt wurde. Meist gehören diese Sprachen zu bestimmten Programmpaketen oder zu ganz bestimmten Aufgaben.
Da wäre AUTOLISP (AutoCAD), das spezielle Fähigkeiten zum Konstruieren und Zeichnen besitzt, die Makrosprache von Lotus 1-2-3 oder FRED, die Programmiersprache, die zu FRAMEWORK gehört. Etwas näher möchten wir hier noch auf dBase eingehen.

dBase

dBase geht auf ein Datenverwaltungsprogramm namens VULCAN zurück, das Ende der siebziger Jahre für CP/M-Systeme entwickelt wurde. Schon die erste Version von dBase enthielt eine Abfrage- und Entwicklungssprache. Diese Sprache ähnelt stark Pascal, enthält aber viele und mächtige Befehle zur Verwaltung von Datenbeständen. Was bei anderen Sprachen viel Arbeit und hunderte von Zeilen Source-Code bedeutet, kann in dBase mit wenigen Zeilen erledigt werden.
Die neueste Version, dBase IV, enthält etwa 400 Befehle und Funktionen, die den Aufbau und die Verwaltung von Daten sehr erleichtern. Neu hinzugekommen sind zahlreiche Möglichkeiten zur gefälligen Gestaltung von Menüs und Windows. Diese Leistungsfähigkeit in bezug auf ein Gebiet fordert natürlich auch Opfer. Nicht datenbankspezifische Probleme lassen sich, wenn überhaupt, nur sehr schwer lösen.

Sprachendschungel

Wie unser kleiner Überblick der Programmiersprachen zeigt, existiert für nahezu jede Problematik eine passende Programmiersprache. Welche nun der einzelne benutzen soll, muß ihm selbst überlassen werden. Ein Programmierer sollte sich in jedem Fall darüber im klaren sein, daß nicht die Wahl der Sprache, sondern die Güte der Algorithmen über die Klasse eines Programms entscheidet.
Da außerdem die meisten Sprachen mehr oder weniger häufig auftretende Parallelen besitzen, sollte keiner große Angst vor dem Betreten dieses Sprachendschungels haben.

(Robert Haas/jf)



Stein auf Stein

Hervorragende Umsetzung des Spiels Tetris

Vor nicht allzu langer Zeit gab es ein Spiel aus Rußland, das einen Riesenerfolg im Computerbereich erlangte. Eine herrliche Adaption dieses Spiels können wir Ihnen und ihrem CPC bieten. Das Programm funktioniert genauso wie sein Vorbild. Des weiteren wurden einige neue Features mit integriert. Doch lassen auch Sie sich überraschen und begeistern.

Um bei CONNECT – so heißt das Spiel – bestehen zu können, muß man dem Spiel seine ungeteilte Aufmerksamkeit und volle Konzentration zuwenden. Es handelt sich um eine Art Puzzlespiel, indem von der oberen Spielfeldbegrenzung die Puzzlesteine herunterfallen. Es fällt jeweils ein Stein, wobei es sieben Möglichkeiten (also sieben verschiedene Steine) gibt. Ist der Stein am unteren Bildschirmrand eingesetzt, folgt der nächste von oben. Dies geht solange weiter, bis der 'Steineberg' den oberen Spielfeldrand erreicht hat. Dann ist das Spiel zu Ende, und daher sollte man vermeiden, den Steineberg zu schnell wachsen zu lassen.

Verbindet man nun durch geschicktes Manövrieren der Steine den linken Spielfeldrand mit dem rechten, so wird die so erzeugte Steinreihe nach unten aus dem Spielfeld geschoben. Die Anzahl der so erreichten Linien wird in der rechten mittleren Hälfte zusammen mit der aktuellen Punktezahl dargestellt.

Eine Statistik gibt Auskunft

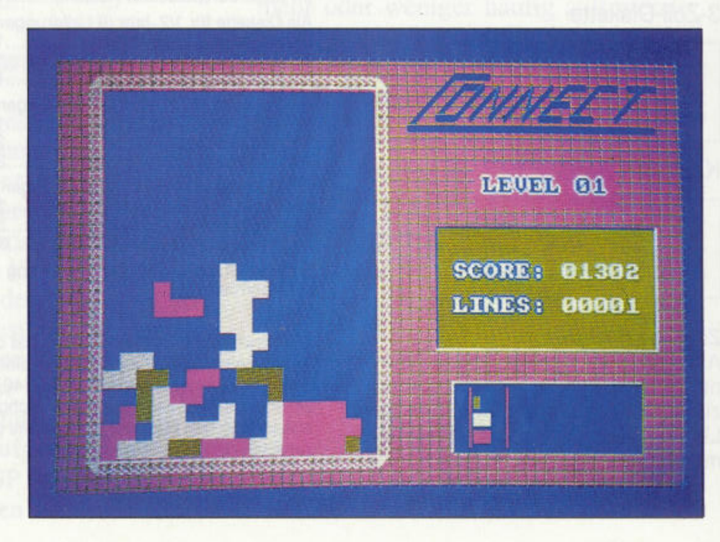
Unter diesem Fenster befindet sich die Anzeige, die die Anzahl der bisher gefallenen Steine in Form von Balken aufzeichnet. Es gibt drei Farbmöglichkeiten für die Steine und daher auch drei Balken. Sind alle drei Balken an der Markierung angekommen, gelangt der Spieler in den nächsten Level, und die bisher erreichten Linien werden als Punkte angerechnet. Mit zunehmendem Level wird die Fallgeschwindigkeit der Steine erhöht, was ihre Manövrierzeitspanne stark verkürzt.

Ebenso erscheinen bei erhöhter Schwierigkeit einteilige Zusatzsteine, die einen Buchstaben tragen. Ein 'P' bedeutet 100 Bonuspunkte, während durch ein 'K' oder ein 'C' 100 Punkte abgezogen werden. Ein 'K' setzt zusätzlich die bisher erreichte Linienanzahl wieder auf Null, während ein 'C' den aktuellen Stein "nur" löscht. Die Aufnahme eines solchen Extrasteins wird durch ein kurzes Bildschirmaufflackern bestätigt. Gesteuert werden die Steine entweder mit den Cursortasten oder dem Joystick. Die Wahl kann am Spielanfang getroffen werden. Durch Drücken des Feuerknopfes kann die Stellung des gerade fallenden Steines verändert werden.

Auf die richtige Stellung kommt es an

Jeder Stein hat vier Stellungen, wobei jede eine andere Einpassungsmöglichkeit bietet. Die Stellung kann man aber nur solange verändern, bis die obere Spitze des Steinbergs erreicht ist. Von nun an kann man den Stein nur noch nach links oder rechts steuern. Befindet sich der Stein direkt an der äußeren Begrenzung des Spielfeldes, kann der Stein trotz des Feuers nicht gedreht werden.

Die Punktevergabe hinsichtlich des Einsetzens richtet sich danach, wie schnell ein Stein eingepaßt ist. Fällt er langsam nach unten, gibt es wenige Punkte, und wenn man den Joystick nach hinten zieht bzw. die entsprechende Cursortaste drückt, fällt der Stein sofort nach unten, und man erhält mehr



CONNECT ist der neueste Knüller auf dem CPC. Aber passen Sie auf, dieses Spiel macht süchtig

Punkte. Dies gelingt nur bei blitzschnellem Erfassen der Situation, da die Fallgeschwindigkeit kein langes Nachdenken zuläßt. Begleitet wird das Spiel von einer Hintergrundmelodie, die man auf Wunsch mit der Taste 'S' an oder ausschalten kann. Auch ist eine Unterbrechung des Spiels möglich. Dazu muß die Taste 'P' gedrückt werden. Ist das Spiel zu Ende, also hat der Steineberg die obere Spielfeldumrandung erreicht, wird die 'Hall of Fame' von CONNECT gezeigt, und der Spieler kann sich eintragen, falls er die entsprechende Punktezahl erreicht hat. Jetzt kann er ein neues Spiel beginnen oder das Programm abbrechen.

(Thorsten Mertsching/cd)

für 464-664-6128

```
10 ' ***************************          [1051]
20 ' ***     CONNECT  V2.1      ***       [846]
30 ' *** Thorsten Mertsching ***          [2812]
40 ' ***  Giessen/April '89  ***          [2256]
50 ' ***************************          [1051]
60 '                                      [117]
70 ' *** TITELBILD & INIT ***             [577]
80 '                                      [117]
90 DIM TON(22),DAU(22):SPEED INK 10,10:ENT [4023]
 -1,2,2,1,4,-2,1,2,2,1
100 ENV 1,=9,4000:ENV 2,2,3,1,6,-1,4:ENV 3 [1999]
,2,8,1,12,-1,2
110 UBG(2)=1.7:FOR I=3 TO 5:UBG(I)=2:NEXT  [1900]
120 RESTORE 3120:FOR I=1 TO 10:READ HOFN$( [3072]
I):HOFS(I)=50000-I*4300:NEXT
130 FOR I=1 TO 6:YBS(I,1)=0:YBS(I,2)=1:YBS [2805]
(I,3)=0:YBS(I,4)=1:NEXT
140 FOR I=3 TO 6:YBS(I,3)=1:NEXT:RANDOMIZE [4899]
 TIME:TSP(1)=1:TSP(3)=1
150 GOSUB 1340:MODE 1:PAPER 0:INK 1,16:INK [3019]
 2,26:INK 3,9:INK 0,2:BORDER 0
160 FOR I=0 TO 12 STEP 4:J=J+1:PLOT I,I,J: [2196]
DRAW I,399-I:DRAW 639-I,399-I
170 DRAW 639-I,I:DRAW I,I:NEXT:TITEL=-1:FO [6760]
R I=0 TO 1:WINDOW#1,2+I,6-I,3+I,11-I
180 WINDOW#2,35+I,39-I,3+I,11-I:PAPER#1,3+ [5583]
I:PAPER#2,3+I:CLS#1:CLS#2:NEXT
190 M$="THOM SOFTWARE PRESENTS":X=9:Y=3:P1 [2881]
=3:P2=1:GOSUB 2440
200 ORIGIN -455,-295:D=0:RESTORE 3050:GOSU [2544]
B 2120:D=0:I=1:J=0
210 FOR V=1 TO 15:FOR C=200-V*10 TO 20 STE [4644]
P -5:SOUND 134,C,1,15:NEXT C,V
220 RESTORE 2700:EVERY 15,1 GOSUB 2640:ORI [3257]
GIN 0,0
230 M$="WRITTEN MARCH'89 BY TH. MERTSCHING [4108]
":P1=2:P2=3:X=3:Y=14:GOSUB 2440
240 M$=CHR$(164)+" 1989 BY PC AMSTRAD INTE [4506]
RNATIONAL":X=3:Y=17:GOSUB 2440
250 M$="PRESS FIRE FOR JOYSTICK GAME":P1=1 [2783]
:P2=3:X=6:Y=21:GOSUB 2440
260 M$="OR COPY FOR KEYBOARD GAME":X=8:Y=2 [5857]
3:GOSUB 2440:TITEL=0:J=0:RESTORE 3240
270 GOSUB 3170:IF NOT INKEY(76) THEN LEFT= [4991]
74:RIGHT=75:BACKW=73:FIRE=76:GOTO 320
280 IF NOT INKEY(9) THEN LEFT=8:RIGHT=1:BA [3888]
CKW=2:FIRE=9 ELSE 270
290 '                                      [117]
300 ' *** SPIELFELDAUFBAU ***              [1412]
310 '                                      [117]
320 ZJ=0:CALL &BCA7:OUT &BC00,1:FOR I=40 T [3159]
O 0 STEP -1:OUT &BD00,I
330 FOR J=1 TO 35:NEXT J,I:GOSUB 3350:EVER [2941]
Y 18,1 GOSUB 3360:CLS
340 INK 1,8:INK 2,8:INK 3,8:PEN 1:FOR I=0  [3022]
TO 2:SRNZ(I)=418:NEXT
350 PLOT 1,1,3:DRAW 1,399:DRAW 639,399:DRA [3705]
W 639,1:DRAW 1,1
360 PRINT CHR$(22)CHR$(1):FOR I=4 TO 644 S [4389]
TEP 10:PLOT I,0,2:DRAW I,400:NEXT
370 FOR I=4 TO 404 STEP 10:PLOT 0,I:DRAW 6 [3425]
40,I:NEXT:ORIGIN 0,0,424,566,298,260
380 CLG 0:ORIGIN 0,0,0,640,0,400:SCORE=0:L [3128]
INES=0:LEV=1
390 LOCATE 3,2:PRINT CHR$(169)STRING$(17,1 [3192]
67)CHR$(170):FOR I=3 TO 23
400 LOCATE 3,I:PRINT CHR$(165):LOCATE 21,I [2605]
:PRINT CHR$(166)
```

Listing CONNECT

```
410 NEXT:PRINT TAB(3)CHR$(171)STRING$(17,1 [2629]
68)CHR$(172):PEN 2
420 PRINT CHR$(22)CHR$(0);:WINDOW#1,4,20,3 [4343]
,23:PAPER#1,3:CLS#1:YSP=23:H=2
430 WINDOW#2,25,38,11,17:PAPER#2,2:CLS#2:W [5235]
INDOW#3,26,37,20,23:PAPER#3,3:CLS#3
440 PLOT 382,127,3:DRAW 382,240:DRAW 608,2 [3431]
40:DRAW 608,127,1:DRAW 382,127
450 PLOT 398,30:DRAW 398,96:DRAW 592,96:DR [3151]
AW 592,30,2:DRAW 398,30
460 M$="LEVEL 0"+MID$(STR$(LEV),2,2):X=27: [5050]
Y=8:P1=1:P2=3:GOSUB 2440
470 PAPER 2:P1=3:P2=1:M$="SCORE:":X=25:Y=1 [3009]
3:GOSUB 2440
480 PRINT" 00000":M$="LINES:":Y=15:GOSUB 2 [3251]
440:PRINT" 00000":PAPER 0
490 RESTORE 3050:GOSUB 2120:INK 0,8:BORDER [3176]
 5:SND=-1:A=REMAIN(1)
500 OUT &BC00,1:FOR I=0 TO 40:IF I=2 THEN  [3423]
INK 1,26:INK 2,12:INK 3,2
510 OUT &BD00,I:FOR J=1 TO 35:NEXT J,I     [1504]
520 RESTORE 1950:EVERY 10,2 GOSUB 1900     [2651]
530 '                                      [117]
540 ' *** HAUPTSCHLEIFE ***                [1012]
550 '                                      [117]
560 X=11:Y=3:XA=11:YA=2:XK=0:YK=3:CST=0:TA [2689]
=0
570 XK=INT(RND*16+1):CW=INT(RND*3+1):IF XK [9513]
>6 AND XK<10 THEN 570 ELSE IF RND*LEV>UBG(
LEV) AND LEV>1 THEN PEN 1:PAPER 3:LOCATE X
K+4,3:PRINT CHR$(172+CW):PAPER 0:CST=-1 EL
SE XK=0
580 STEIN=INT(RND*7)+1:STELL=1:PE=INT(RND* [3168]
3)
590 LOCATE X,Y:PEN PE:PRINT SHAPE$(STEIN,S [1544]
TELL)
600 IF YSP>4 THEN 630 ELSE A=REMAIN(2):WIN [3190]
DOW#1,7,17,11,13
610 FOR I=300 TO 3200 STEP 10:SOUND 2,I,1, [3248]
15:NEXT:PEN#1,2:PAPER#1,0:X=7:Y=12
620 PRINT#1,STRING$(33,207);:M$="GAME OVER [3159]
":P1=2:P2=3:GOSUB 2440:GOTO 1500
630 GOSUB 2390:GOSUB 1170:WINDOW#1,4,20,3, [6619]
YSP:EVERY 20-LEV*2,1 GOSUB 1230
640 EL=0:TM=INT(RND*30)+80:IF DONE=3 THEN  [2080]
2220
650 XA1=4:XA2=18:IF STEIN=1 AND STELL/2=IN [2972]
T(STELL/2) THEN XA2=20:GOTO 670
660 IF STEIN=7 OR STELL/2=INT(STELL/2) THE [1918]
N XA2=19
670 ALT=STELL                              [567]
680 IF NOT INKEY(LEFT) AND X>XA1 THEN XA=X [2592]
:X=X-1:GOTO 1290
690 IF NOT INKEY(RIGHT) AND X<XA2 THEN XA= [2230]
X:X=X+1:GOTO 1290
700 IF NOT INKEY(BACKW) THEN Y=YSP-3:T=0:G [1887]
OTO 740
710 IF NOT INKEY(FIRE) AND STEIN<7 AND Y+3 [2831]
<YSP AND X>XA1 AND X<XA2 THEN 1280
720 XA=X:IF CST AND TEST((XK+3)*16,400-YK* [3073]
16)<>3 THEN GOSUB 860
730 IF Y+3<YSP THEN 650                    [1226]
740 EL=REMAIN(1):YT=Y+1:YPLT=Y+B:GOSUB 100 [2210]
0
750 XA=X:IF T>0 THEN 800                   [1833]
760 IF NOT INKEY(LEFT) AND TEST((X-1)*16-4 [3376]
,404-(Y+LG(STEIN,STELL))*16)=3 THEN XA=X:X
=X-1
770 IF NOT INKEY(RIGHT) AND TEST((X+STX(ST [6375]
EIN,STELL)-1)*16+4,404-(Y+LG(STEIN,STELL))
*16)=3 THEN XA=X:X=X+1
780 LOCATE XA,YA+1:PEN 3:PRINT SHAPE$(STEI [4272]
N,STELL):PEN PE:YA=Y:Y=Y+1
```

Listing CONNECT

```
790 LOCATE X,Y:PRINT SHAPE$(STEIN,STELL):Y [5626]
T=Y+1:GOSUB 1000:GOTO 750
800 IF YT-LG(STEIN,STELL)<YSP THEN YSP=YT- [3479]
LG(STEIN,STELL)
810 IF XK>0 THEN LOCATE XK+4,YK:PEN 3:PRIN [3469]
T CHR$(143)
820 GOSUB 940:GOSUB 1080:GOTO 560          [3625]
830 '                                       [117]
840 ' *** EXTRA-STEINE ***                 [2307]
850 '                                       [117]
860 CST=0:IF CW>1 THEN 880                  [979]
870 TM=100:GOSUB 940:INK 1,0:FOR I=1 TO 15 [3900]
:NEXT:INK 1,26:RETURN
880 INK 2,26:FOR I=1 TO 15:NEXT:INK 2,12:T [2616]
M=-100:GOSUB 940:IF CW>2 THEN 900
890 LINES=0:LOCATE 33,15:PEN 1:PAPER 2:PRI [5906]
NT"00000":PAPER 0:RETURN
900 A=REMAIN(1):LOCATE XA,YA+1:PEN 3:PRINT [5458]
 SHAPE$(STEIN,STELL):GOTO 560
910 '                                       [117]
920 ' *** PUNKTE ERHOEHEN ***              [1261]
930 '                                       [117]
940 SCORE=SCORE+TM:M$=STR$(SCORE):M$=MID$( [2906]
M$,2,LEN(M$)-1)
950 M$=STRING$(5-LEN(M$),"0")+M$:PAPER 2:P [2169]
EN 1:LOCATE 33,13:PRINT M$
960 PAPER 0:RETURN                         [1143]
970 '                                       [117]
980 ' *** SITZT STEIN FEST ? ***            [728]
990 '                                       [117]
1000 T=0:YTK=YT+YBS(STEIN,STELL):FOR I=X+X [5050]
SR(STEIN,STELL) TO X+XSP(STEIN,STELL)
1010 IF TEST(I*16-4,392-YTK*16)<>3 THEN TA [2052]
=-1:T=T+1
1020 NEXT:IF ZPX(STEIN,STELL)>0 AND T=0 TH [6739]
EN IF TEST((X+ZPX(STEIN,STELL)-1)*16-4,392
-(Y+ZPY(STEIN,STELL)-2)*16)<>3 THEN T=T+1
1030 IF T=0 AND STEIN>2 AND STEIN<6 AND ST [5804]
ELL=3 THEN IF TEST((X+TSP(STEIN-2))*16,400
-(Y+2)*16)<>3 THEN T=T+1
1040 RETURN                                 [555]
1050 '                                      [117]
1060 ' *** BEIDE SEITEN VERBUNDEN ? ***    [2140]
1070 '                                      [117]
1080 YT=YT-1:FOR IL=-1 TO 2:PRUEF=0:YPR=39 [5258]
2-(YT+IL)*16:IF YT+IL>22 THEN 1130
1090 FOR I=55 TO 320 STEP 16:IF TEST(I,YPR [2958]
)<>3 THEN PRUEF=PRUEF+1
1100 NEXT I:IF PRUEF<>17 THEN 1130 ELSE WI [3910]
NDOW#1,4,20,3,YT+IL+1:PRINT#1,CHR$(11)
1110 LINES=LINES+1:M$=STR$(LINES):M$=MID$( [5030]
M$,2,LEN(M$)-1):M$=STRING$(5-LEN(M$),"0")+
M$
1120 PAPER 2:PEN 1:LOCATE 33,15:PRINT M$:P [3825]
RUEF=0:PAPER 0:YSP=YSP+1
1130 NEXT IL:RETURN                        [1284]
1140 '                                      [117]
1150 ' *** ENDE LEVEL ? ***                [1096]
1160 '                                      [117]
1170 DONE=0:XST=SRNZ(PE)+2:IF XST<ENDP+2 T [6081]
HEN A=PE+1:YST=(25+A*16):PLOT XST,YST,PE:D
RAW XST,YST+10:SRNZ(PE)=XST
1180 FOR I=0 TO 2:IF SRNZ(I)=ENDP THEN DON [3060]
E=DONE+1
1190 NEXT:RETURN                            [940]
1200 '                                      [117]
1210 ' *** INTERRUPT STEINSCROLLING ***    [2683]
1220 '                                      [117]
1230 IF NOT INKEY(BACKW) THEN RETURN       [1642]
1240 YA=Y:Y=Y+1:LOCATE#1,1,1:PRINT#1,CHR$( [4237]
11):TM=TM-3:YK=YK+1:RETURN
1250 '                                      [117]
1260 ' *** STEIN DREHEN ***                [1105]
1270 '                                      [117]
1280 STELL=STELL+1:IF STELL=5 THEN STELL=1 [3934]
1290 DI:LOCATE XA,YA+1:PEN 3:CALL &BD19:PR [4584]
INT SHAPE$(STEIN,ALT)
1300 LOCATE X,Y:PEN PE:CALL &BD19:PRINT SH [4291]
APE$(STEIN,STELL):EI:GOTO 720
1310 '                                      [117]
1320 ' *** NEUE ZEICHEN ***                [1077]
1330 '                                      [117]
1340 SYMBOL AFTER 164                      [1452]
1350 SYMBOL 165,34,74,86,34,34,74,86,34    [1617]
```

Listing CONNECT

```
1360 SYMBOL 166,68,106,82,68,68,106,82,68  [2253]
1370 SYMBOL 167,0,102,153,68,34,68,255,0   [1805]
1380 SYMBOL 168,0,255,34,68,34,153,102,0   [1806]
1390 SYMBOL 169,0,2,7,14,30,62,127,34      [1878]
1400 SYMBOL 170,0,64,224,112,120,124,254,6 [2637]
8
1410 SYMBOL 171,34,127,62,30,14,7,2,0      [1929]
1420 SYMBOL 172,68,254,124,120,112,224,64, [2809]
0
1430 SYMBOL 173,62,65,93,85,93,81,65,62    [1729]
1440 SYMBOL 174,62,65,85,89,89,85,65,62    [1941]
1450 SYMBOL 175,62,65,93,81,81,93,65,62    [1902]
1460 RETURN                                 [555]
1470 '                                      [117]
1480 ' *** SPIELENDE > HALL OF FAME ***    [2194]
1490 '                                      [117]
1500 RESTORE 1850:FOR I=1 TO 2000:NEXT:OUT [1972]
 &BC00,1
1510 FOR I=40 TO 0 STEP -1:OUT &BD00,I:FOR [3337]
 j=1 TO 35:NEXT J,I
1520 EVERY 8,1 GOSUB 1830:FOR I=1 TO 10:IF [2766]
 SCORE<HOFS(I) THEN NEXT
1530 K=I:IF K=11 THEN 1560                 [1257]
1540 FOR J=9 TO K STEP -1:HOFS(J+1)=HOFS(J [4789]
):HOFN$(J+1)=HOFN$(J):NEXT
1550 HOFN$(K)="":HOFS(K)=SCORE             [1960]
1560 MODE 1:PEN 3:PAPER 2:FOR I=1 TO 5:PRI [2903]
NT"CONNECT*";:NEXT:PAPER 0
1570 FOR i=16 TO 382 STEP 4:PLOT 0,i,RND*2 [4366]
:DRAW 640,i:NEXT
1580 PAPER 3:WINDOW 1,40,2,6:CLS:PEN 2:PAP [5663]
ER 2:WINDOW 14,27,3,5:CLS:PAPER 3
1590 WINDOW 3,8,9,22:CLS:WINDOW 33,38,9,22 [4642]
:CLS:PAPER#1,2:WINDOW#1,12,29,10,21
1600 CLS#1:PLOT 175,62,3:DRAW 175,257:DRAW [4125]
 465,257:DRAW 465,62,1:DRAW 175,62
1610 WINDOW 1,40,1,25:A$=STRING$(10,32):LO [3073]
CATE 1,7:PRINT A$
1620 LOCATE 1,24:PRINT A$:LOCATE 31,7:PRIN [4067]
T A$:LOCATE 31,24:PRINT A$
1630 A$=STRING$(8,32):LOCATE 2,8:PRINT A$: [3082]
LOCATE 2,23
1640 PRINT A$:LOCATE 32,8:PRINT A$:LOCATE  [2834]
32,23:PRINT A$
1650 PLOT 205,316,0:DRAW 205,370:DRAW 435, [2785]
370:DRAW 435,316:DRAW 205,316
1660 PAPER 0:WINDOW 1,40,1,25:M$="HALL OF  [3672]
FAME":X=14:Y=4:P1=1:P2=1
1670 GOSUB 2440:PRINT#1:FOR I=1 TO 10:PEN# [2943]
1,1
1680 PRINT#1,TAB(2);HOFN$(I);TAB(13);:IF I [2904]
=K THEN PEN#1,0
1690 PRINT#1,USING"#####";HOFS(I):NEXT:WHI [3479]
LE INKEY$<>"":WEND:INK 0,5:BORDER 7
1700 OUT &BC00,1:FOR I=0 TO 40:OUT &BD00,i [4491]
:FOR j=1 TO 35:NEXT j,i:A=REMAIN(1)
1710 IF K=11 THEN 1740 ELSE EVERY 20,1 GOS [4246]
UB 2600:LOCATE#1,2,1+K:A$="":B$=""
1720 PEN#1,0:GOSUB 2540:IF B$="" THEN B$=" [4692]
ANONYM":PRINT#1,B$ ELSE PRINT#1," "
1730 HOFN$(K)=B$:A=REMAIN(1)               [1563]
1740 PEN 1:A$="FIRE = NEW GAME      ANY OTH [2336]
ER KEY = QUIT"
1750 WHILE INKEY$<>"":WEND:RESTORE 3310    [2382]
1760 Y$=INKEY$:IF NOT INKEY(FIRE) THEN 320 [3485]
1770 IF Y$<>"" THEN CALL &BCA7:CLS:CALL &B [2696]
C02:PEN 1:END
1780 GOSUB 3290:I=I+1:IF INT(I/2)=I/2 THEN [3515]
 A$=LOWER$(A$) ELSE A$=UPPER$(A$)
1790 PEN 1:LOCATE 1,25:PRINT A$;:GOTO 1760 [1924]
1800 '                                      [117]
1810 ' *** SOUND: AUFBAUPAUSE 1 ***        [1506]
1820 '                                      [117]
1830 READ A:IF A=999 THEN A=REMAIN(1):RETU [2651]
RN
1840 SOUND 1,A,15,15:SOUND 2,A+1,15,15:RET [2173]
URN
1850 DATA 100,90,100,110,120,110,100,0,130 [2503]
,120,110,0,120,110,100,0
1860 DATA 100,90,100,110,120,110,100,0,130 [2622]
,0,100,0,120,150,999
1870 '                                      [117]
1880 ' *** SOUND: SPIEL ***                [1258]
1890 '                                      [117]
```

Listing CONNECT

```
1900 IF NOT INKEY(60) THEN IF SND THEN SND [3362]
=0 ELSE SND=-1
1910 IF NOT INKEY(27) THEN 2020            [1270]
1920 READ TON:IF TON=999 THEN RESTORE 1950 [2368]
:GOTO 1920
1930 IF SND THEN SOUND 2,TON,20,,1:SOUND 1 [2518]
,TON+2,20,,1
1940 RETURN                                 [555]
1950 DATA 338,319,284,253,253,284,253,190, [2042]
213,253,284,319
1960 DATA 379,379,319,253,239,213,213,190, [2465]
213,253,319,253,284,284
1970 DATA 284,284,338,319,284,253,284,253, [3270]
190,213,253,284,319,379
1980 DATA 379,338,319,284,253,239,253,284, [2697]
319,284,319,319,213,999
1990 '                                      [117]
2000 ' *** SPIELPAUSE ***                  [1619]
2010 '                                      [117]
2020 A$="":A=REMAIN(1):A=REMAIN(2):WHILE I [5101]
NKEY$<>"":WEND
2030 WINDOW#6,25,38,11,11:PAPER#6,2:PEN#6, [2691]
0:FOR I=1 TO 3
2040 A$=A$+"GAME PAUSED * ":NEXT:SOUND 1,2 [2313]
0,1,15:J=0
2050 CALL &BD19:IF INKEY$<>"" THEN 2070 EL [3293]
SE J=J+1:IF J=28 THEN J=1
2060 CALL &BD19:PRINT#6,MID$(A$,J,14);:FOR [2272]
 I=1 TO 50:NEXT:GOTO 2050
2070 SOUND 1,10,1,15:CLS#6:IF EL=0 THEN EV [4415]
ERY 20-LEV*2,1 GOSUB 1230
2080 EVERY 10,2 GOSUB 1900:RETURN          [2054]
2090 '                                      [117]
2100 ' *** NAMENSZUG ***                   [1192]
2110 '                                      [117]
2120 READ A,B,C:IF A=999 THEN RESTORE 3050 [4881]
:G=0:D=D+2:IF D=8 THEN D=0:RETURN ELSE 212
0
2130 G=G+1:IF TITEL THEN SOUND 2,200-(D+G) [2233]
*3,1,15
2140 FK=1.6:IF D<6 THEN H=3 ELSE H=2       [2050]
2150 IF TITEL THEN IF A=1 THEN PLOT B*FK,( [5725]
C+D)*FK,H ELSE DRAW B*FK,(C+D)*FK,H
2160 IF TITEL THEN 2120                    [1020]
2170 IF A=1 THEN PLOT B,C+D,H ELSE DRAW B, [2137]
C+D,H
2180 GOTO 2120                              [307]
2190 '                                      [117]
2200 ' *** NAECHSTER LEVEL ***             [1647]
2210 '                                      [117]
2220 A=REMAIN(1):A=REMAIN(2):CLS#2:PEN#2,1 [2398]
2230 PRINT#2,"YOU 'VE PASSED  LEVEL 0"+MI [3069]
D$(STR$(LEV),2,2)
2240 FOR I=1 TO 15:FOR J=300-I*10 TO 90 ST [2708]
EP -10:SOUND 2,J,1,15:NEXT J,I
2250 J(1)=2:J(2)=1:GOSUB 3440:PEN#2,0:PRIN [4247]
T#2:PRINT#2,"*BONUS**BONUS*":PEN#2,3
2260 INK 0,8,16:FOR I=LINES TO 0 STEP -1:L [3765]
OCATE#2,1,5:PRINT#2,USING"##";I;
2270 BON=LEV*30:SOUND 1,40+I,1,15:PRINT#2, [3237]
"LINES x";BON;
2280 SCORE=SCORE+BON:PRINT#2,"   =";SCORE  [1638]
2290 FOR K=1 TO 150:NEXT K,I:LEV=LEV+1:INK [4308]
 0,8:IF LEV>5 THEN LEV=1
2300 CALL &BCA7:SOUND 1,600,300,15:SOUND 2 [4101]
,604,300,15:FOR I=1 TO 2500:NEXT
2310 CLS#2:CLS#3:ORIGIN 0,0,520,565,298,26 [3593]
0:CLG 0:ORIGIN 0,0,0,640,0,400
2320 M$="0"+MID$(STR$(LEV),2,2):X=33:Y=8:P [3600]
1=1:P2=3:GOSUB 2440
2330 P1=3:P2=1:M$="SCORE:":X=25:Y=13:GOSUB [4149]
 2440:TM=0:GOSUB 940:M$="LINES:":Y=15
2340 GOSUB 2440:PAPER 2:PRINT" 00000":PAPE [4360]
R 0:j(1)=3:j(2)=3:GOSUB 3440
2350 LINES=0:FOR I=0 TO 2:SRNZ(I)=418:NEXT [3042]
:YSP=23:GOTO 520
2360 '                                      [117]
2370 ' *** MARKIERUNG SETZEN ***           [2200]
2380 '                                      [117]
2390 PLOT 416,90,0:DRAW 416,36:ENDP=420+30 [4104]
*LEV:PLOT ENDP+4,90:DRAW ENDP+4,36
2400 RETURN                                 [555]
2410 '                                      [117]
2420 ' *** STRINGAUSGABE ***                [808]
```

Listing CONNECT

```
2430 '                                         [117]
2440 PLOT -2,0,P1:X1=X-1:Y1=Y-1:FOR Q=1 TO [5993]
 LEN(M$):PRINT CHR$(23)CHR$(3)
2450 TAG:M1$=MID$(M$,Q,1):X1=X1+1          [1821]
2460 MOVE X1*16+2,398-Y1*16:PRINT M1$;:MOV [3367]
E X1*16,400-Y1*16:PRINT M1$;
2470 MOVE X1*16,396-Y1*16:PRINT M1$;:MOVE  [3188]
X1*16-2,398-Y1*16:PRINT M1$;
2480 TAGOFF:PRINT CHR$(23)CHR$(0);:PEN P2: [5751]
LOCATE X+Q,Y:PRINT CHR$(22)CHR$(1);
2490 IF TITEL THEN SOUND 2,Q*10,1,15,,,3   [1336]
2500 PRINT M1$;CHR$(22)CHR$(0);:NEXT:RETUR [1821]
N
2510 '                                     [117]
2520 ' *** EINGABE-ROUTINE ***             [1336]
2530 '                                     [117]
2540 A$=UPPER$(INKEY$):PRINT#1," "CHR$(8); [2628]
"*"CHR$(8);
2550 IF A$="" THEN 2540 ELSE A=ASC(A$)     [2128]
2560 IF A=127 AND LEN(B$)>0 THEN PRINT#1,C [4032]
HR$(8);"* "CHR$(8)CHR$(8);:B$=LEFT$(B$,LEN
(B$)-1)
2570 IF A=13 THEN RETURN ELSE IF LEN(B$)=9 [3558]
 THEN 2540
2580 IF A<32 OR A>90 THEN 2540             [1593]
2590 PRINT#1,A$"*"CHR$(8);:B$=B$+A$:GOTO 2 [1817]
540
2600 SOUND 1,10,2,15:RETURN                [2149]
2610 '                                     [117]
2620 ' *** DATENBLOECKE EINLESEN ***       [2569]
2630 '                                     [117]
2640 IF J=4 THEN J=0:I=I+1:IF I>7 THEN HAL [3353]
T=-1:A=REMAIN(1):RETURN
2650 J=J+1:READ LAE:FOR K=1 TO LAE         [2170]
2660 READ WERT:SHAPE$(I,J)=SHAPE$(I,J)+CHR [3555]
$(WERT):NEXT K
2670 READ XSR(I,J),XSP(I,J),STX(I,J),LG(I, [3973]
J),ZPX(I,J),ZPY(I,J)
2680 RETURN                                [555]
2690 ' STEIN 1                             [615]
2700 DATA 4,10,143,143,143,0,2,3,2,0,0     [1609]
2710 DATA 7,143,10,8,143,10,8,143,0,0,1,3, [2227]
0,0
2720 DATA 4,10,143,143,143,0,2,3,2,0,0     [1609]
2730 DATA 7,143,10,8,143,10,8,143,0,0,1,3, [2227]
0,0
2740 ' STEIN 2                             [610]
2750 DATA 9,9,143,143,8,8,8,10,143,143,0,1 [2259]
,3,2,3,2
2760 DATA 8,143,10,8,143,143,10,8,143,1,1, [2465]
2,3,1,3
2770 DATA 9,9,143,143,8,8,8,10,143,143,0,1 [2259]
,3,2,3,2
2780 DATA 8,143,10,8,143,143,10,8,143,1,1, [2465]
2,3,1,3
2790 ' STEIN 3                             [613]
2800 DATA 8,9,143,10,8,8,143,143,143,0,2,3 [1418]
,2,0,0
2810 DATA 9,143,10,8,143,143,8,8,10,143,0, [2306]
0,2,3,2,3
2820 DATA 8,10,143,143,143,8,8,10,143,1,1, [2177]
3,2,1,3
2830 DATA 10,9,143,8,8,10,143,143,10,8,143 [2060]
,1,1,2,3,1,3
2840 ' STEIN 4                             [624]
2850 DATA 6,143,8,10,143,143,143,0,2,3,2,0 [1388]
,0
2860 DATA 9,143,143,10,8,8,143,10,8,143,0, [1394]
0,2,3,2,2
2870 DATA 7,10,143,143,143,10,8,143,2,2,3, [1342]
2,1,3
2880 DATA 10,9,143,10,8,143,10,8,8,143,143 [3257]
,0,1,2,3,0,0
2890 ' STEIN 5                             [627]
2900 DATA 10,9,9,143,8,8,8,10,143,143,143, [2326]
0,2,3,2,0,0
2910 DATA 8,143,10,8,143,10,8,143,143,0,1, [1701]
2,3,0,0
2920 DATA 9,10,143,143,143,10,8,8,8,143,0, [2068]
0,3,2,2,3
2930 DATA 8,143,143,10,8,143,10,8,143,1,1, [1265]
2,3,1,2
2940 ' STEIN 6                             [622]
2950 DATA 6,143,143,10,8,143,143,1,2,3,2,1 [1435]
,2
```

Listing CONNECT

```
2960 DATA 11,9,143,10,8,8,143,143,10,8,8,1 [1586]
43,0,0,2,3,2,3
2970 DATA 6,143,143,10,8,143,143,1,2,3,2,1 [1435]
,2
2980 DATA 11,9,143,10,8,8,143,143,10,8,8,1 [1586]
43,0,0,2,3,2,3
2990 ' STEIN 7                             [625]
3000 DATA 7,143,143,10,8,8,143,143,0,1,2,2 [2073]
,0,0
3010 DATA 7,143,143,10,8,8,143,143,0,1,2,2 [2073]
,0,0
3020 DATA 7,143,143,10,8,8,143,143,0,1,2,2 [2073]
,0,0
3030 DATA 7,143,143,10,8,8,143,143,0,1,2,2 [2073]
,0,0
3040 ' SCHRIFTDATEN                        [1251]
3050 DATA 1,610,330,2,355,330,2,405,380,2, [2730]
440,380,1,385,340,2,415,370
3060 DATA 2,435,370,2,405,340,2,385,340,1, [2311]
420,340,2,450,370,2,440,340
3070 DATA 2,470,370,1,456,340,2,486,370,2, [2843]
476,340,2,506,370,1,492,340
3080 DATA 2,522,370,2,540,370,1,507,355,2, [3493]
522,355,1,492,340,2,514,340
3090 DATA 1,546,340,2,524,340,2,554,370,2, [3417]
572,370,1,585,370,2,612,370
3100 DATA 1,570,340,2,600,370,999,0,0      [1678]
3110 ' HALL OF FAME                        [836]
3120 DATA THE BEST,INGENIUS,GREAT!,SUPER,V [4396]
ERY GOOD,OKAY
3130 DATA NOT BAD,BE BETTER,FORGET IT,LOSE [1141]
R
3140 '                                     [117]
3150 ' *** TITELANIMATION ***              [2472]
3160 '                                     [117]
3170 J=J+1:IF J=9 THEN J=1 ELSE LOCATE#1,1 [3629]
,1:PRINT#1,CHR$(8):LOCATE#2,1,1:PRINT#2,CH
R$(8)
3180 IF J=1 THEN PEN#1,INT(RND*2+1):PEN#2, [9404]
INT(RND*2+1):LOCATE#1,1,1:PRINT#1,SHAPE$(I
NT(RND*7+1),INT(RND*4+1));:LOCATE#2,1,1:PR
INT#2,SHAPE$(INT(RND*7+1),INT(RND*4+1));
3190 '                                     [117]
3200 ' *** SOUND: TITELMELODIE ***         [847]
3210 '                                     [117]
3220 READ T:IF T=999 THEN RESTORE 3240:RET [1781]
URN
3230 SOUND 1,T+4,25,5,1,1:SOUND 2,T,25,5,1 [3937]
,1:SOUND 132,T,25,5,1,1:RETURN
3240 DATA 239,179,179,142,159,179,159,142, [2346]
179,119,106,119,179
3250 DATA 106,179,119,119,142,159,179,179, [2817]
142,179,239,119,999
3260 '                                     [117]
3270 ' *** SOUND: HALL OF FAME ***         [720]
3280 '                                     [117]
3290 READ a:IF a=999 THEN RESTORE 3310:GOT [2781]
O 3290
3300 SOUND 7,0,20,7,2,a:RETURN             [1423]
3310 DATA 20,30,5,20,10,30,10,20,30,10,10, [2098]
20,999
3320 '                                     [117]
3330 ' *** SOUND: AUFBAUPAUSE 2 ***        [1419]
3340 '                                     [117]
3350 RESTORE 3380:FOR I=1 TO 22:READ TON(I [3724]
),DAU(I):NEXT:RETURN
3360 ZJ=ZJ+1:IF ZJ>22 THEN A=REMAIN(1):RET [3829]
URN
3370 SOUND 2,TON(ZJ),DAU(ZJ),,1,1:RETURN   [3522]
3380 DATA 319,50,253,75,239,25,253,25,284, [2590]
40,253,50,239,25,253,25
3390 DATA 284,25,253,40,319,35,284,50,358, [2794]
25,319,50,379,25,358,25
3400 DATA 379,40,426,25,478,50,319,50,1046 [2649]
,50,1396,75,999,0
3410 '                                     [117]
3420 ' *** SPIELFELD LOESCHEN ***          [1180]
3430 '                                     [117]
3440 J=1:FOR I=32 TO 366 STEP 4:GOSUB 3460 [3846]
:NEXT
3450 J=2:FOR I=367 TO 32 STEP -4:GOSUB 346 [3640]
0:NEXT:RETURN
3460 PLOT 48,I,J(J):DRAW 318,I:SOUND 2,0,1 [2206]
,5,,,10:RETURN
```

Listing CONNECT

Laurel II

Die Hilfe für Maschinencode-Programme

Dieses Programm ist entstanden, als der Autor (wieder einmal) vor der Situation stand, eine Maschinenroutine zur Unterstützung eines BASIC-Programms zu schreiben. Viele kleine Routinen, die schon vorhanden waren, mußten geändert, zusammengebunden und neu assembliert werden; eine zeitraubende und frustrierende Arbeit. Diese Arbeit übernimmt nun LAUREL II ebenso wie die Anlage von RSXen. Man kann sich nun eine Bibliothek von kleinen MC-Modulen einrichten, die nach Bedarf in der richtigen Reihenfolge zu einem Macro zusammengestellt werden.

Leistungsmerkmale

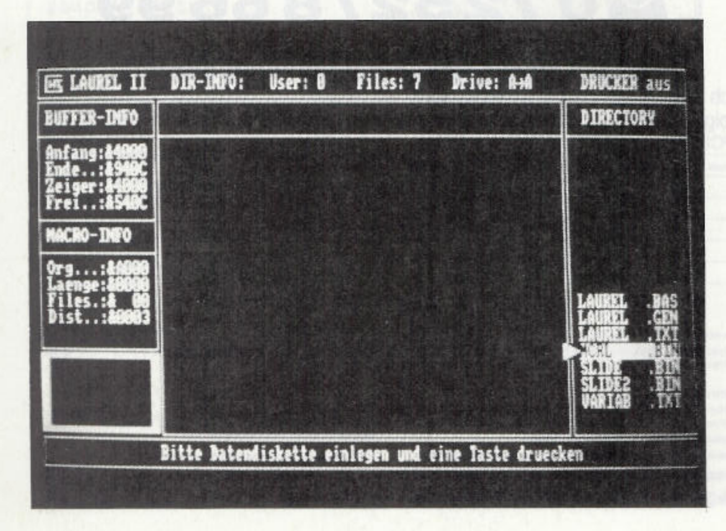
Das Utility LAUREL II kann folgendes leisten:

1. Verschieben von Maschinenprogrammen an eine beliebige Adresse im gesamten RAM-Bereich mit automatischer Korrektur der Adressen und Sprünge
2. Linken (verbinden) von bis zu 32 (Voreinstellung, ggf.mehr) MC-Modulen zu einem Makro bis zu 21 kByte Länge (ggf. mehr)
3. Komfortable Verwaltung der Disketten (Rename,Erase)
4. Optionales Erstellen von RSX-Befehlen für jedes Modul
5. LAUREL II arbeitet auf allen USER-Ebenen und mit allen Laufwerkskombinationen (AMSDOS).
6. Freie Wahl der Distanzen zwischen den einzelnen Modulen
7. Ausführliche Dokumentation aller Ergebnisse auf dem Bildschirm und – wenn ein Drucker online ist – zusätzlich auch auf diesem
8. Alle Operationen sind speicherplatzsparend und geschwindigkeitsoptimiert. Das ist durchaus von Belang, denn für das Verschieben von nur 1 kByte MC-Code sind zirka 35000 Vergleichsoperationen nötig (Zeitaufwand in BASIC: ca. 2 Min./LAUREL II: 3 Sek.).
9. Weitgehende Absicherung des Programmes gegen Fehler und Fehleingaben. (Ausnahme: CPC 464)

Wichtiger Hinweis: Wie alle MC-Manipulatoren hat LAUREL II Schwierigkeiten mit Listen. Wenn Daten einer Liste zufällig dem Format eines Sprunges (z.B. JP XXXX) oder einer Wertübergabe (z.B. LD HL,XXXX) entsprechen und die Änderungskriterien erfüllt sind, wird diese Änderung ausgeführt. Es wird empfohlen, diese Listen genau zu überwachen.

Hinweise zum Abtippen

Die REM-Zeilen werden nicht angesprungen, können also weggelassen werden. Es wird empfohlen, die Zeilenorganisation so zu belassen, wie sie ist. Die ERROR-Routinen erfordern eine bestimmte Konfiguration. Es sind zwei Listings abzutippen. Das erste ist das Hauptprogramm (BASIC-Listing), das zweite Listing (Datalader) enthält Maschinencode und ist unter laurel2 .bas abzuspeichern. Danach kann es mit RUN gestartet werden.



Das Menü von Laurel II

Bedienung des Programms

1. Bildschirm: Der Bildschirm ist in verschiedene Fenster aufgeteilt:
obere Zeile: Dir-Info, Druckerstatus
links oben: Bufferinformationen
links Mitte: Macroinformationen
links unten: Fehlerinformationen
rechts: Directory (Endlosschleife)
untere Zeile: Menüs und Anweisungen (Laufschrift)
Mitte: File-Infos und Eingaben (ggf. + Drucker)

Eingaben (allgemein)

Leereingaben führen in das jeweils nächsthöhere Menü oder sichern den alten Wert.

2.1. Menüs:
Eingabe des jeweils großgeschriebenen Buchstabens bzw. einer Zahl führen zur gewählten Routine.

Leer- oder Falscheingaben führen zurück ins Menü.

2.2. Directory:
Das Directory wird in einer Endlosschleife an einer markierten Stelle vorbeigerollt. Die Markierung oder Demarkierung erfolgt dort durch 'SPACE', das Scrollen mit 'CURSOR UP' und der Abschluß mit 'ENTER' oder 'RETURN'. (Anmerkung: Der normale Diskettenkatalog wird nicht angezeigt! Ausnahme: Rename und Erase).

2.3. Zahlen:
Eingabe erfolgt grundsätzlich hexadezimal (ohne "&"). Nur Distanzen werden dezimal eingegeben.

2.4. Namen:
Bei 'Rename','Execute' und 'Save' werden File- oder (optional) RSX-Namen angefordert. Diese sollten den üblichen Konventionen entsprechen.

Nach Angabe von RSX-Namen werden bei Anwahl von 'Save' automatisch RSXen für die entsprechenden Module angelegt. Die Routine für die RSXen wird an das Makro angehängt. Die Einsprungadresse des Makros zeigt dann auf die RSX-Routine.

2.5. Diskettenwechsel:
ist möglich **vor** oder **nach** der 'Execute'- Routine, in jedem Fall **vor** 'Save'. Nach 'Save' würde ein neues File angelegt, das natürlich direkt an das alte anschließen kann. RSXen können jedoch nur für ein Makro angelegt werden.

Arbeitsweise des Programms

Katalog: Hier werden die äußeren Bedingungen der Arbeit festgelegt und gegebenenfalls die Diskette mit 'Rename' oder 'Erase' reorganisiert.
Reset: Alle Programm-Pointer werden auf die Startwerte zurückgesetzt. Der Buffer wird **nicht** gelöscht.
Makro: Hauptteil des Programms
– Org: Hier wird die Adresse für das 1. Byte des zu erzeugenden Makros gefordert. Eingabe: beliebig. Default: &A000.
Es empfiehlt sich **nicht**, einen einmal benutzten Org-Wert zu ändern.
– Distanz: Vor jedem Modul wird nach dem Abstand zum nächsten gefragt. Defaultwert ist 3 (reicht für einen JUMP, möglich sind alle Werte 0-...
– Save: Abschluß des Programms. Dabei wird (Idee von G. Woigk) ein Binärdatei-Header gegebenenfalls die RSX-Routine erzeugt und die Daten des Buffers sequentiell unter dem angegebenen Makronamen auf Diskette abgelegt.
– Execute: 1. Auswahl der zu linkenden Files über die DIR-Routine (invers markieren), die Reihenfolge ist bindend.
2. Überprüfung, ob die gewählten Files tatsächlich Binärfiles sind (wenn nicht – Abbruch und Rücksprung).
3. Drucker ein/aus
4. Die Programme werden in der gewählten Reihenfolge hintereinander in den Buffer geladen und dort vom Relocator so bearbeitet, als stünden sie an der richtigne Adresse (org bzw.ng).

Variable	Bedeutung
an$(x)	= Anweisungen
m$(x),mi$(x)	= Menuestrings
t$(x),tx$,bi$	= Titel oder Strings für Print Using
f$,f$(x)	= Filenamen
us	= USER
in$,aus$	= Laufwerk
st(x)	= Startadresse der Files
rsx$	= RSX-Namen
rsx	= Anzahl der angemeldeten RSXen
ba,be,pt	= Buffer: Anfang,Ende,Pointer
org	= ORG
ng	= neuer File-ORG
disb	= Distanzbytes
login	= MC-Routine LOG-IN
kat	= " " Katalog
mswp	= " " swap CAT-Buffer nach f$
vswp	= " " Variablen-swap
lauf	= " " Laufschrift
reloc	= " " Relocator
dr$(x)	= Drucker ein/aus
pa,pe,lg	= Originaladressen Anfang,Ende,Länge der Files
dis	= Distanz zwischen Originaladresse und ORG
dis1	= " " Buffer und Org
norg,point	= Übergabeadressen an reloc
flag	= ERROR-Flag
rflag	= RSX-Flag
ab	= Wert für SPC(ab)
anz	= Anzahl der markierten Files
df	= " " Files auf der Diskette

Die Variablen-Liste von Laurel II

Wichtiger Hinweis

Dieses Programm ist nicht für den absoluten Anfänger geeignet, da einiges an Grundwissen vorausgesetzt wird.

(Wolf-Thomas Gerner/cd)

für 464-664-6128



```
5 'LAUREL II * 8/89 v.3 * Wolf-Thomas Gern [2010]
er
10 CLEAR:CALL &BB4E:CALL &BC02:ON ERROR GO [3089]
TO 7000
15 SYMBOL AFTER 252                        [1396]
97 '                                        [117]
98 '***** MASKE *****                       [730]
99 '                                        [117]
100 SYMBOL 252 ,&F0,&FF,&FF,&FF,&FF,&FF,&F [2908]
F,&F0
110 SYMBOL 253 ,&0,&0,&F0,&FF,&FF,&F0,&0,& [2322]
0
120 SYMBOL 254 ,&FF,&C0,&CC,&CC,&CF,&CC,&C [1973]
0,&FF
130 SYMBOL 255 ,&FF,&0,&FF,&CC,&CC,&CF,&3, [1876]
&FF
140 INK 0,0:INK 1,24:BORDER 0:MODE 2:PEN#6 [1381]
,0
150 FOR i=1 TO 5:READ a,b,c,d              [1426]
160 MOVE a,b:DRAWR c,0:DRAWR 0,-d:DRAWR -c [2409]
,0:DRAWR 0,d
170 NEXT                                    [350]
180 FOR i=1 TO 5:READ a,b,c:MOVE a,b:DRAWR [3164]
 c,0:NEXT
190 ORIGIN 0,0,3,107,116,42:CLG 1           [955]
200 LOCATE 68,2:PRINT"DRUCKER aus"         [1719]
210 LOCATE 68,4:PRINT"DIRECTORY"           [1597]
220 LOCATE 65,18:PRINT CHR$(252)CHR$(253)  [1238]
230 LOCATE 2,2:PRINT CHR$(254)CHR$(255)" L [3294]
AUREL II"
```

Listing LAUREL

```
240 b$="....:&":b=3:GOSUB 290:b=10:GOSUB 2 [2059]
90
250 FOR i=0 TO 6:READ l,r,o,u:WINDOW#i,l,r [3508]
,o,u:NEXT:CLS#6
260 GOSUB 400:GOSUB 2500:GOSUB 2505        [2205]
270 GOTO 700                                [425]
280 '                                       [117]
290 FOR i=1 TO 6:READ a$:LOCATE 2,b+i:PRIN [3467]
T a$;
300 IF a$<>"" AND i>2 THEN PRINT RIGHT$(b$ [2272]
,8-LEN(a$))
310 NEXT:RETURN                             [940]
397 '                                       [117]
398 '**** VAR ****                          [547]
399 '                                       [117]
400 DEF FNa(a)=PEEK(a)+256*PEEK(a+1):DEF F [2412]
Nh(a)=255 AND INT(a/256):DEF FNl(a)=255 AN
D UNT(a)
410 DIM f$(32),t$(11),st(32),rsx$(32)       [948]
415 bi$="\  \\  \\  \\  \":mi$="\  \\  \   [2525]
\\\  \"
420 FOR i=1 TO 32:f$(i)=STRING$(12,"*"):NE [3918]
XT:f$=STRING$(12,"*"):l$="*"
430 login=&A410:kat=&A413:mswp=&A416:vswp= [2414]
&A419:lauf=&A41C:reloc=&A41F
440 p$=CHR$(1)+CHR$(9):us=0:mf=0:df=0:in$= [3611]
"A":aus$="A"
450 ba=16384:pt=ba:org=40960:disb=3:be=379 [2660]
00
455 norg=&A601:point=&A5FF:dr$(0)="Drucker [2941]
 aus":dr$(1)="Drucker an"
460 an$(1)=SPACE$(12)+"Bitte Datendiskette [6903]
 einlegen und eine Taste druecken"
462 an$(2)=SPACE$(25)+"**** Bitte warten   [3422]
****"
```

Listing LAUREL

```
464 an$(3)=SPACE$(21)+"**** Bitte eine Tas [3520]
te druecken ****"
466 an$(4)=SPACE$(15)+"Bitte Wert eingeben [4550]
 / nur ENTER = alter Wert"
468 an$(5)=SPACE$(13)+"Bitte RSX-Namen ein [5251]
geben  / nur ENTER = keine RSX"
470 d$="DIR-INFO:    User:##     Files:##    [4428]
 Drive: \ \"
480 m$(0)=SPACE$(22)+"MENUE:    Catalog   M [5034]
acro   Reset":mi$(0)="cmr"
490 m$(1)=SPACE$(18)+"CATALOG:    Drive   U [4421]
ser  Rename   Erase":mi$(1)="dure"
500 m$(2)=SPACE$(17)+"MACRO:    Org    Dista [5020]
nz   Execute   Save":mi$(2)="odes"
510 m$(3)=SPACE$(6)+"RESET:    Achtung!  Al [8992]
le Daten gehen verloren!  Ausfuehren (j/n)
?":mi$(3)="j"
520 m$(4)=SPACE$(14)+"DRIVE:  1 (A"+p$+"A) [4420]
  2 (A"+p$+"B)  3 (B"+p$+"A)  4 (B"+p$+"B)
":mi$(4)="1234"
530 m$(5)=SPACE$(11)+"USER:   0  1  2  3   [5449]
4  5  6  7  8  9  A  B  C  D  E  F":mi$(5)
="0123456789abcdef"
550 t$(2)="Nr.Filename         Adresse   Sta [3117]
rt RSX-Name"
560 tx$=  "##.\          \  \  \-\  \  \    [2444]
\  "
570 tx1$= "              \  \  \-\  \  \    [3227]
\  "
580 t$(6)="R E N A M E":t$(7)="E R A S E": [2844]
t$(8)="O R G":t$(9)="D I S T A N Z"
585 t$(10)="TEST auf BINAERFILE":t$(11)="S [1774]
 A V E"
590 RETURN                                [555]
597 '                                     [117]
598 '**** DATAS ****                      [615]
599 '                                     [117]
600 DATA 1,388,636,26,1,36,636,26,1,358,10 [3264]
9,238,114,358,402,318,520,358
610 DATA 117,318,114,326,402,1,326,109,520 [2655]
,326,117,1,246,109,1,214,109
620 DATA BUFFER-INFO,Anfang,Ende,Zeiger,F [5403]
rei,MACRO-INFO,,Org,Laenge,Files,Dist
630 DATA 17,64,6,22,17,64,4,4,10,13,6,9,10 [4756]
,13,13,17,3,78,24,24,67,78,6,22,2,13,19,22
697 '                                     [117]
698 '***** INIT *****                     [687]
699 '                                     [117]
700 SYMBOL AFTER 256:MEMORY &A40F         [2030]
710 OPENOUT"d":MEMORY HIMEM-1:CLOSEOUT:LOA [4738]
D"mcrl.bin",&A410:MEMORY &3FFF
997 '                                     [117]
998 '***** Hauptschleife *****            [1091]
999 '                                     [117]
1000 m=0:GOSUB 2000:IF i=0 THEN 1000      [1268]
1005 m=i                                  [192]
1010 CLS:CLS#1:GOSUB 2000:IF i0 THEN 1000 [1264]
1020 mn=i:ON m GOTO 1100,1300,1750        [1206]
1030 GOTO 1010                            [345]
1097 '                                    [117]
1098 '*** Catalog ***                     [937]
1099 '                                    [117]
1100 aa=m:ON mn GOTO 1120,1150,1240,1220  [2636]
1117 '                                    [117]
1118 '*** Drive                           [1119]
1119 '                                    [117]
1120 m=4:GOSUB 2000:IF i THEN in$=CHR$(65- [3599]
(i>2)):aus$=CHR$(65-(i=2 OR i=4))
1130 GOTO 1180                            [399]
1147 '                                    [117]
1148 '*** User                            [418]
1149 '                                    [117]
1150 m=5:GOSUB 2000:IF i THEN us=i-1      [1692]
1180 m=aa:GOSUB 2500:GOTO 1010            [1507]
1215 '                                    [117]
1218 '*** Erase                           [689]
1219 '                                    [117]
1220 ab=18:m=7:GOSUB 2520:GOSUB 2030:GOSUB [3185]
 2100
1221 FOR i=1 TO anz                       [856]
1222 PRINT f$(i)" ist geloescht":|ERA,@f$( [3746]
i)
1224 NEXT:CALL login:df=PEEK(&A422):GOTO 1 [3232]
180
```

Listing LAUREL

```
1237 '                                    [117]
1238 '*** Rename                          [762]
1239 '                                    [117]
1240 ab=17:m=6:GOSUB 2520:GOSUB 2030:GOSUB [4191]
 2100:r=4:GOSUB 2027
1241 FOR i=1 TO anz                       [856]
1242 PRINT"Alter Name: "f$(i);:INPUT" - Ne [2472]
u: ",a$
1244 IF a$="" THEN 1255 ELSE |REN,@a$,@f$(i [2126]
)
1255 NEXT:m=aa:GOTO 1010                  [793]
1297 '                                    [117]
1298 '*** Macro ****                      [1216]
1299 '                                    [117]
1300 aa=m:ON mn GOTO 1350,1370,1400,1800  [2907]
1347 '                                    [117]
1348 ' *** Org                            [429]
1349 '                                    [117]
1350 ab=20:m=8:GOSUB 2520:r=4:GOSUB 2027  [1661]
1355 PRINT"Organisation alt: ab &"HEX$(org [5278]
,4):PRINT TAB(14)"neu: ab &";
1360 INPUT"",a$:IF a$<>""THEN org=VAL("&"+ [5362]
a$):org=org-65536*(org<0)
1365 m=aa:GOSUB 2510:GOTO 1010            [1356]
1367 '                                    [117]
1368 '*** Distanz                         [1870]
1369 '                                    [117]
1370 ab=18:m=9:GOSUB 2520                 [1081]
1380 r=4:GOSUB 2027                       [739]
1382 PRINT"Anzahl Distanzbytes (NOP) alt:" [5345]
disb:PRINT TAB(27)"neu:";
1385 INPUT"",a$:IF a$<>""THEN disb=VAL(a$) [2801]
:GOSUB 2510
1387 IF ex THEN RETURN                    [699]
1390 GOTO 1365                            [406]
1397 '                                    [117]
1398 '*** Execute                         [976]
1399 '                                    [117]
1400 ex=-1:GOSUB 2030:IF anz=0 THEN 1500  [1628]
1405 GOSUB 2100:GOSUB 2800:IF jj THEN 1500 [1877]
1410 GOSUB 2080:ab=0:m=2:GOSUB 2520:GOSUB [3572]
2505
1415 disl=org-ba                          [864]
1417 '                                    [117]
1420 FOR f=1 TO anz                       [901]
1425 GOSUB 1380:PRINT:pt=pt+disb:GOSUB 250 [2213]
5
1430 LOAD f$(f),pt                        [823]
1440 pa=FNa(&A76A):pl=FNa(&A76D):pe=pa+pl- [2504]
1:dis=org-pa
1445 IF pt+pl>be THEN ERROR 42            [1547]
1447 IF flag THEN f=anz:GOTO 1480         [1303]
1450 st=FNa(&A76F):ng=pt+disl:GOSUB 2505:r [4159]
=5:GOSUB 2027:GOSUB 2530
1460 r=2:GOSUB 2027:GOSUB 1700:pt=pt+pl:GO [2245]
SUB 2505
1480 NEXT:f=f-1:IF flag=0 THEN r=3:GOSUB 2 [3961]
027:CALL &BB06
1500 flag=0:ex=0:CLS#6:GOTO 1365          [1684]
1697 '                                    [117]
1698 '*** Relocator                       [667]
1699 '                                    [117]
1700 POKE norg,FNl(ng):POKE norg+1,FNh(ng) [2310]
1710 POKE point,FNl(pt):POKE point+1,FNh(p [2719]
t)
1720 CALL reloc                           [105]
1730 RETURN                               [555]
1747 '                                    [117]
1748 '*** Reset                           [959]
1749 '                                    [117]
1750 pt=ba:org=40960:disb=3:f=0:rsx=0:us=0 [2884]
:df=0:in$="A":aus$="A"
1760 rflag=0:f=0:anz=0:GOSUB 2505:GOSUB 25 [3760]
00:GOTO 1000
1797 '                                    [117]
1798 '*** Save                            [534]
1799 '                                    [117]
1800 IF pt=ba THEN ERROR 43               [680]
1802 IF flag THEN flag=0:GOTO 1365        [1730]
1804 ab=18:m=11:GOSUB 2520:r=1:GOSUB 2027: [1896]
CALL &BB06
1805 PRINT:INPUT"Name des Macros: ",a$:IF [2449]
a$=""THEN 1365
```

Listing LAUREL

```
1807 r=2:GOSUB 2027:IF rflag=0 THEN GOSUB  [2117]
2600
1810 ¦DRIVE,aus$:OPENOUT a$                 [2376]
1815 POKE &A7B1,2                           [613]
1820 POKE &A7B4,FNl(org):POKE &A7B5,FNh(or [1828]
g)
1830 POKE &A7B9,FNl(st):POKE &A7BA,FNh(st) [2197]
1840 ll=pt-ba:POKE &A7B7,FNl(ll):POKE &A7B [2936]
8,FNh(ll)
1850 FOR i=ba TO pt:PRINT#9,CHR$(PEEK(i)); [3304]
:NEXT
1860 CLOSEOUT                               [902]
1870 GOSUB 2560                             [827]
1890 r=3:GOSUB 2027:CALL &BB06:GOTO 1365    [1836]
1997 '                                      [117]
1998 '***** SUBS *****                      [675]
1999 '                                      [117]
2000 :CALL vswp,@m$(m),@l$:GOSUB 2020:CALL [3941]
 vswp,@m$(m),@l$
2010 w$="":WHILE w$="":w$=INKEY$:WEND:i=IN [3457]
STR(mi$(m),LOWER$(w$)):RETURN
2015 '                                      [117]
2020 :WINDOW SWAP 0,7:CALL lauf,@l$:WINDOW [2659]
 SWAP 0,7:RETURN
2025 '                                      [117]
2027 CALL vswp,@an$(r),@l$:GOSUB 2020:CALL [3806]
 vswp,@an$(r),@l$:RETURN
2029 '                                      [117]
2030 r=1:GOSUB 2027:CALL &BB06              [848]
2040 ¦DRIVE,in$:¦USER,us:ad=&BB5A:b=PEEK(a [3178]
d):POKE ad,&C9:CAT
2045 POKE ad,b:IF flag THEN flag=0:RETURN  [2898]
2050 CALL login:df=PEEK(&A422):IF df=0 THE [1699]
N ERROR 40
2055 IF flag THEN flag=0:RETURN            [1345]
2060 GOSUB 2500:WINDOW SWAP 0,5:CALL kat:C [3986]
LS:WINDOW SWAP 0,5:anz=PEEK(&A425)
2070 RETURN                                 [555]
2079 '                                      [117]
2080 PAPER#6,1:CLS#6:PRINT#6,"* VORSICHT * [6391]
"SPC(2)"Diskette"SPC(3)"n i c h t"SPC(4)"w
echseln":PAPER#6,0:RETURN
2089 '                                      [117]
2100 FOR i=1 TO anz:CALL mswp,@f$(i):NEXT: [2975]
RETURN
2109 '                                      [117]
2499 '                                      [117]
2500 LOCATE#7,17,2:PRINT#7,USING d$;us,df; [4252]
in$+p$+aus$:RETURN
2501 '                                      [117]
2505 LOCATE#2,1,1:PRINT#2,USING bi$;HEX$(b [5257]
a,4),HEX$(be,4),HEX$(pt,4),HEX$(be-pt,4);
2510 LOCATE#3,1,1:PRINT#3,USING mi$;HEX$(o [5134]
rg,4),HEX$(pt-ba,4),HEX$(f,2),HEX$(disb,4)
;:RETURN
2519 '                                      [117]
2520 CLS#1:PRINT#1,SPC(ab)t$(m);:RETURN    [2162]
2527 '                                      [117]
2528 '*** Files ausgeben u. RSX anmelden   [3182]
2529 '                                      [117]
2530 x=0:GOSUB 2533:IF (INP(&F500)AND 64)= [2641]
0 THEN x=8 ELSE RETURN
2533 PRINT#x,USING tx$;f,f$(f),HEX$(pa,4), [4004]
HEX$(pe,4),HEX$(st,4)
2535 PRINT#x,USING tx1$;f$(f),HEX$(ng,4),H [3432]
EX$(ng+pl-1,4),HEX$(ng+st-pa,4);
2540 IF x=8 THEN PRINT#x,"¦"rsx$(rsx):GOTO [3113]
 2550
2545 INPUT"¦",a$:IF a$<>""THEN rsx=rsx+1:r [3696]
sx$(rsx)=UPPER$(a$):st(rsx)=ng+st-pa
2550 PRINT#x:RETURN                         [1326]
2559 '                                      [117]
2560 x=0:GOSUB 2565:IF (INP(&F500)AND 64)= [3071]
0 THEN x=8 ELSE RETURN
2565 PRINT#x:PRINT#x,"Macro: "a$           [1753]
2570 PRINT#x,"ORG: &"HEX$(org,4)" Laenge:  [4032]
&"HEX$(ll,4)" START: &"HEX$(st,4)
2580 IF rsx THEN PRINT#x:FOR i=1 TO rsx:PR [3933]
INT#x,"¦"rsx$(i):NEXT
2590 RETURN                                 [555]
2597 '                                      [117]
2598 '*** RSX einrichten                    [1813]
2599 '                                      [117]
2600 IF rsx=0 THEN st=org:RETURN ELSE rfla [2616]
g=-1
```

Listing LAUREL

```
2605 PRINT:PRINT"RSXen werden eingerichtet [4362]
."
2610 RESTORE 2750:st(rsx+1)=pt+dis1:FOR i= [2624]
1 TO rsx
2615 c$=rsx$(i):a=@c$:lr=PEEK(a):a=a+1:b=F [1892]
Na(a)
2620 x=pt+lr:WHILE pt<x:POKE pt,PEEK(b):pt [5201]
=pt+1:b=b+1:WEND
2630 POKE pt-1,(PEEK(b-1)OR &80)           [934]
2640 NEXT:POKE pt,0:pt=pt+1:st(rsx+2)=pt+d [3427]
is1
2650 k=rsx+1:GOSUB 2730                    [1534]
2660 READ a:FOR k=1 TO rsx:GOSUB 2725:NEXT [4188]
:st(rsx+3)=pt+dis1:pt=pt+4:st=pt+dis1
2670 k=rsx+2:GOSUB 2720                    [749]
2680 k=rsx+3:GOSUB 2720                    [1419]
2690 FOR i=1 TO 4:READ a:POKE pt,a:pt=pt+1 [3623]
:NEXT
2700 RETURN                                [555]
2720 READ a                                [428]
2725 POKE pt,a:pt=pt+1                     [592]
2730 POKE pt,FNl(st(k)):pt=pt+1:POKE pt,FN [3595]
h(st(k)):pt=pt+1:RETURN
2750 DATA &c3,1,&21,&cd,&d1,&bc,&c9        [1342]
2797 '                                     [117]
2798 '*** Test auf Binaer                  [725]
2799 '                                     [117]
2800 ab=13:m=10:GOSUB 2520:r=2:GOSUB 2027  [2627]
2810 jj=0:FOR i=1 TO anz:PRINT f$(i);      [1762]
2820 OPENIN f$(i):CLOSEIN                  [1413]
2830 j=PEEK(&A767)<>2:IF j THEN jj=j:PRINT [4037]
" - KEIN Binaer-File":GOTO 2850
2840 PRINT" - o.k."                        [916]
2850 NEXT:IF jj THEN ERROR 41              [1597]
2860 IF jj THEN flag=0:anz=0 ELSE r=3:GOSU [2993]
B 2027:GOSUB 2870
2865 CLS:RETURN                            [832]
2867 '                                     [117]
2868 '*** Druckerabfrage                   [2220]
2869 '                                     [117]
2870 w$="":WHILE w$="":w$=INKEY$:dr=(INP(& [2106]
F500) AND 64)<>0:dr=dr+1
2880 LOCATE#7,68,2:dd=NOT dd:IF dd THEN PR [4779]
INT#7,dr$(dr)ELSE PRINT#7,SPACE$(11)
2890 WEND:LOCATE#7,68,2:PRINT#7,dr$(dr):CL [2673]
S:RETURN
6997 '                                     [117]
6998 '*** Fehler ***                       [1262]
6999 '                                     [117]
7000 IF ERR=25 THEN RESUME NEXT            [1726]
7002 POKE &BB5A,207:WINDOW SWAP 0,6:PAPER  [2393]
1
7003 CLS:PRINT" -Fehler-":PRINT            [1360]
7010 IF ERR=32 THEN 7400                   [1113]
7020 IF ERR>39 THEN ON ERR-39 GOTO 7200,72 [2035]
10,7220,7230
7190 STOP                                  [464]
7200 PRINT"Keine Datei";:PRINT"auf USER"us [4792]
;:GOTO 7500
7210 PRINT"* ABBRUCH *":GOTO 7500          [802]
7220 PRINT"Buffer":PRINT"ist voll":GOTO 75 [2723]
00
7230 PRINT"Buffer":PRINT"ist leer":GOTO 75 [2857]
00
7400 PRINT"Disk-":PRINT"ERROR              [2208]
7500 PAPER 0:WINDOW SWAP 0,6:flag=-1:r=3:G [4580]
OSUB 2027:CALL &BB06:CLS#6:RESUME NEXT
```

Listing LAUREL

```
1000 MEMORY &A40F                          [722]
1010 FOR i=1 TO 736                        [554]
1020 READ a$:POKE HIMEM+i,VAL("&"+a$)      [1124]
1030 NEXT:SAVE"mcrl.bin",b,&A410,736       [2562]
1040 DATA C3,26,A4,C3,3F,A4,C3,32          [1688]
1050 DATA A5,C3,77,A5,C3,94,A5,C3          [843]
1060 DATA 26,A6,00,00,00,00,97,32          [847]
1070 DATA 24,A4,06,00,2A,9B,A7,3E          [1091]
1080 DATA 00,BE,28,06,04,CD,66,A4          [909]
1090 DATA 18,F5,78,32,22,A4,C9,CD          [1574]
1100 DATA 26,A4,97,32,23,A4,21,12          [1459]
1110 DATA 01,CD,75,BB,01,00,05,2A          [1213]
```

Listing LAUREL

```
1120 DATA 9B,A7,3A,22,A4,3D,B9,0C          [955]
1130 DATA 38,13,CD,7A,A4,CD,66,A4          [1569]
1140 DATA 10,F0,C3,B4,A4,C9,B7,11          [872]
1150 DATA 0E,00,ED,5A,C9,21,12,01          [1652]
1160 DATA CD,75,BB,3E,20,CD,5A,BB          [1178]
1170 DATA 18,E6,C5,E5,7E,FE,FF,C4          [1167]
1180 DATA A0,A4,06,08,CD,AE,A4,10          [963]
1190 DATA FB,3E,2E,CD,5A,BB,06,03          [1648]
1200 DATA CD,AE,A4,10,FB,3A,23,A4          [939]
1210 DATA FE,00,C4,A0,A4,E1,C1,C9          [1326]
1220 DATA 3A,23,A4,EE,FF,32,23,A4          [881]
1230 DATA 3E,18,CD,5A,BB,C9,23,7E          [1558]
1240 DATA CD,5A,BB,C9,06,01,2A,9B          [1182]
1250 DATA A7,CD,06,BB,FE,F0,28,0C          [1455]
1260 DATA FE,20,CC,FE,A4,FE,0D,CA          [1229]
1270 DATA 17,A5,18,ED,3A,22,A4,B8          [833]
1280 DATA CA,42,A4,04,CD,66,A4,E5          [880]
1290 DATA 21,12,01,CD,75,BB,E1,3A          [1555]
1300 DATA 22,A4,FE,06,38,03,D6,04          [806]
1310 DATA B8,3E,20,DC,5A,BB,DA,B9          [1542]
1320 DATA A4,E5,11,38,00,ED,5A,CD          [1316]
1330 DATA 7A,A4,E1,C3,B9,A4,E5,21          [2076]
1340 DATA 0D,01,CD,75,BB,E1,3E,FF          [2000]
1350 DATA BE,20,07,3A,24,A4,3C,32          [1427]
1360 DATA 24,A4,77,CD,7A,A4,C9,3A          [936]
1370 DATA 22,A4,47,0E,00,2A,9B,A7          [1084]
1380 DATA 3E,FF,BE,28,01,0C,CD,66          [965]
1390 DATA A4,10,F5,79,32,25,A4,C3          [463]
1400 DATA 65,A4,DD,66,01,DD,6E,00          [869]
1410 DATA 23,7E,5F,23,7E,57,D5,3A          [1393]
1420 DATA 24,A4,47,97,C5,3C,F5,2A          [1992]
1430 DATA 9B,A7,3A,22,A4,47,F1,BE          [856]
1440 DATA 28,0A,CD,66,A4,10,F8,C1          [1678]
1450 DATA 10,EA,D1,C9,C1,3E,FF,77          [1161]
1460 DATA 23,D1,06,08,7E,12,23,13          [1241]
1470 DATA 10,FA,3E,2E,12,13,06,03          [573]
1480 DATA 7E,12,23,13,10,FA,C9,DD          [1632]
1490 DATA 66,01,DD,6E,00,DD,56,03          [1597]
1500 DATA DD,5E,02,06,03,CD,8B,A5          [770]
1510 DATA 10,FB,C9,1A,F5,7E,12,F1          [1551]
1520 DATA 77,23,13,C9,C3,9A,A5,00          [1409]
1530 DATA 00,00,26,03,2E,17,CD,1A          [1641]
1540 DATA BC,22,97,A5,1A,47,3E,4B          [1511]
1550 DATA 90,32,99,A5,13,1A,6F,13          [710]
1560 DATA 1A,67,C5,E5,CD,D9,A5,E1          [2467]
1570 DATA 7E,CD,5A,BB,23,E5,CD,E4          [2058]
1580 DATA A5,E1,C1,10,ED,3A,99,A5          [1466]
1590 DATA 47,C5,CD,D9,A5,3E,20,CD          [1249]
1600 DATA 5A,BB,CD,E4,A5,C1,10,F1          [2246]
1610 DATA C9,26,4E,2E,18,CD,75,BB          [1442]
1620 DATA CD,19,BD,C9,2A,97,A5,06          [1565]
1630 DATA 08,C5,E5,54,5D,1B,01,4C          [1058]
1640 DATA 00,ED,B0,E1,11,00,08,ED          [1254]
1650 DATA 5A,C1,10,ED,C9,00,00,00          [1702]
1660 DATA 00,00,00,00,00,00,00,00          [1005]
1670 DATA 00,01,11,21,22,2A,31,32          [1107]
1680 DATA 3A,C2,C3,C4,CA,CC,CD,D2          [985]
1690 DATA D4,DA,DC,E2,E4,EA,EC,F2          [1129]
1700 DATA F4,FA,FC,ED,DD,FD,2A,6A          [1663]
1710 DATA A7,22,05,A6,ED,5B,6D,A7          [1512]
1720 DATA ED,53,03,A6,B7,ED,5A,22          [1458]
1730 DATA 07,A6,2A,01,A6,ED,5B,05          [1389]
1740 DATA A6,B7,ED,52,22,FD,A5,21          [629]
1750 DATA 09,A6,06,1D,C5,E5,7E,2A          [739]
1760 DATA FF,A5,ED,4B,03,A6,BE,F5          [1261]
1770 DATA CC,71,A6,23,0B,78,FE,00          [1769]
1780 DATA 20,0C,79,FE,00,20,07,F1          [618]
1790 DATA E1,23,C1,10,DF,C9,F1,18          [1453]
1800 DATA E5,FE,ED,CA,98,A6,FE,DD          [1834]
1810 DATA CA,83,A6,FE,FD,CA,83,A6          [1148]
1820 DATA C3,C6,A6,23,0B,7E,FE,21          [1508]
1830 DATA CA,C6,A6,FE,22,CA,C6,A6          [1696]
1840 DATA FE,2A,CA,C6,A6,2B,03,C9          [1068]
1850 DATA 23,0B,7E,FE,43,CA,C6,A6          [1083]
1860 DATA FE,53,CA,C6,A6,FE,63,CA          [1641]
1870 DATA C6,A6,FE,73,CA,C6,A6,FE          [879]
1880 DATA 4B,CA,C6,A6,FE,5B,CA,C6          [2343]
1890 DATA A6,FE,6B,CA,C6,A6,FE,7B          [1370]
1900 DATA CA,C6,A6,2B,03,C9,23,7E          [1779]
1910 DATA 5F,23,7E,57,E5,2A,05,A6          [1523]
1920 DATA B7,ED,52,30,08,2A,07,A6          [759]
1930 DATA B7,ED,52,30,04,0B,0B,E1          [1157]
1940 DATA C9,2A,FD,A5,B7,ED,5A,EB          [1366]
1950 DATA E1,72,2B,73,23,E5,18,ED          [1135]
```

Listing LAUREL

AMS-Line

der direkte Draht zur Firma AMSTRAD

Wie jeden Monat präsentieren wir Ihnen die neuesten Informationen der Firma AMSTRAD.

Farbmonitore für PC 2x86

In letzter Zeit werden häufiger Fragen zum Anlaufverhalten unserer Monitore gestellt.

Nach dem Einschalten wird die volle Helligkeit erst nach mehreren Minuten erreicht. Dies ist konstruktionsbedingt und ist darauf zurückzuführen, daß der Heizfaden der Bildröhre mit zunehmender Betriebstemperatur auch mehr Elektronen abgibt.

Bei unseren Geräten wird die Bildröhre schonend beheizt, um die Lebensdauer zu erhöhen, was den beschriebenen Effekt verursacht.

Amstrad in BTX

Seit einiger Zeit ist AMSTRAD auch in BTX vertreten.

Sie erreichen unser BTX-Programm unter 0610520031 oder einfacher unter *Amstrad#.

Unser Programm enthält Informationen über unsere Produkte und neue Aktionen. Selbstverständlich können Sie uns über den BTX-Anschluß auch eine Nachricht schicken.

Viel Spaß beim Blättern in unseren Seiten!

Neuer Maustreiber für PC2286/PC2386

Auf den deutschen Diskettensätzen, die den Geräten inzwischen beiliegen (siehe amsline 12/89), ist unter anderem eine neue Version des Maustreibers MOUSE.COM enthalten. Die Versionsnummer, die beim Aufruf angezeigt wird, lautet 5.0c3. Dieser Treiber wurde in bezug auf die Unterstützung der VGA-Grafikkarte verbessert.

Unter anderem ist durch den neuen Treiber das Problem "Word 4.0 auf VGA-Karte" (amsline 08/89) für die beiden oben genannten Geräte ausgestanden. Word 4.0 oder auch das neue 5.0 läuft mit dem neuen Treiber problemlos – auch unter dem VGA-Modus; es sind keine zusätzlichen Umstellungen mehr notwendig.

Word 5.0 auf unseren Geräten

Neben dem positiven Punkt, daß der Maustreiber problemlos arbeitet, sollen hier kurz noch einige andere Aspekte zu Microsofts neuestem Kind angesprochen werden.

1) Zunächst die positive Seite: Die in der amsline 11/89 angebotenen Druckertreiber für die AMSTRAD LQ-Drucker laufen auch unter der 5.0-Version.

2) Das SETUP-Programm von Word 5.0 ist etwas "eigenwillig". Wenn man es auf den Geräten der 2x86-Serie aufruft, erhält man nur einen schwarzen Bildschirm. Der VGA-geschädigte Word-4.0-Benutzer schaltet hier natürlich gleich: Versuchen wir es doch 'mal unter Hercules. Und siehe da: Nach der Umschaltung mit DEVICE DISPLAY HERC0 läuft das SETUP-Programm einwandfrei. Nachdem dann Word für VGA installiert wurde, ist auch das Arbeiten mit Word problemlos möglich.

3) Beim PC 1640 mit MD-Monitor läßt sich Word 5.0 nicht für die Hercules-Karte installieren. Der geladene Bildschirmtreiber (SCREEN.VID) verursacht ein Durchlaufen des Bildes, was nicht über die Regler nachregulierbar ist. Zur Zeit gibt es leider nur einen Weg, Word 5.0 dennoch auf dieser Gerätekonfiguration zu installieren:

– Schalten Sie den fünften DIP-Schalter am Gerät von ON (unten) auf OFF (oben). Damit befindet sich die Grafikkarte im EGA-Mono-Modus.

– Installieren Sie Word 5.0 über das SETUP-Programm auf EGA-Grafik.

In dieser Konfiguration läuft Word 5.0 – aber leider keines der vorher installierten Programme: das heißt, es muß jedesmal vor bzw. nach dem Benutzen von Word das Gerät ausgeschaltet und die DIP-Schalter-Einstellung verändert werden.

Wegen der beiden letztgenannten Probleme stehen wir in Kontakt mit der Firma Microsoft, damit eine befriedigende Lösung – zum Beispiel Änderung der entsprechenden Treiber – gefunden wird. Sobald diese vorliegt, werden wir Sie an dieser Stelle darüber informieren.

Belegung der Laufwerksbuchse bei den Geräten der Serie 2x86

In der amsline 7/89 veröffentlichten wir die Belegung der Anschlußbuchsen für die externen Laufwerke. In dieser Belegung hat sich leider ein Fehler eingeschlichen. Wir bitten um Entschuldigung und liefern Ihnen hier noch einmal die richtige Belegung:

Spannungsversorgung (Aufsicht):
oben rechts und links: Masse
unten links: +12 V
unten rechts: +5 V

Laufwerkssignale:
1 - 17 Masse
18 frei
19 High Density
20 frei
21 frei
22 Index
23 Drive Select 0
24 Drive Select 1
25 frei
26 Motor On
27 Direction
28 Step
29 Write Data
30 Write Gate
31 Track 0
32 Write Protect
33 Read Data
34 Side Select
35 Disk Change
36 Masse

Dieser Fehler macht sich nur bei HD-Laufwerken bzw. bei Laufwerken, die von der normalen Shugart-Bus-Belegung abweichen, bemerkbar und ist uns aus diesem Grund bei dem ersten Artikel zu diesem Thema entgangen.

Und nun wünscht Ihnen die AMSTRAD-Belegschaft einen guten Rutsch ins neue Jahr.

Ihre

Daniela

Wie lenke ich einen Roboter?

Interface im Selbstbau

Eine Möglichkeit, die viele CPC-Benutzer wahrscheinlich noch gar nicht genutzt haben, ist die Steuerung von Geräten und Apparaturen in der "Außenwelt". Dabei ist dies mit den CPC-Computern recht einfach und ohne großen Aufwand zu bewerkstelligen. Der Druckeranschluß des Computers, häufig auch Centronics-Port genannt, ist der Schlüssel zu dieser "Welt".

An diesem Ausgang liegen beim CPC 7 Bit parallel vor. Das berühmte achte Bit ist nur durch kleine Eingriffe, über die schon viel geschrieben wurde, zu nutzen. Man muß nun natürlich wissen, wo die Signale zur Verfügung stehen. Wenn man den Druckeranschluß von hinten betrachtet, liegt Pin 1 dabei rechts oben. Die Pin-Nummern nehmen von Pin 1 aus von rechts nach links zu. Diese Belegung ist auch im Handbuch des Computers recht gut dargestellt. Nun die Pinbelegung im einzelnen. Pin 1 liefert das Strobbe-Signal. Dies ist für unsere Zwecke nicht interessant. Pin 2 bis 8 sind für uns wichtig, sie entsprechen den Datenleitungen 0-6. Pin 9 ist beim CPC immer auf Null gesetzt. Dieser Pin entspricht dem achten Bit. Pin 11 ist der Busy-Kanal, auf dem der Drucker meldet, daß keine weiteren Daten mehr empfangen werden können. Dieser Pin läßt sich als Eingabekanal nutzen, doch möchte ich darauf jetzt nicht näher eingehen, da meiner Meinung nach der Joystick-Eingang eine einfachere Möglichkeit der Eingabe darstellt, doch dazu später. Die Pins 14 bis 33 liegen alle auf Masse.

Mit dem Befehl OUT &EFFF, <Wert> ist es nun ohne Probleme möglich, Daten zum Centronics-Port zu senden. Der Port behält nun so lange den gesendeten Wert, bis er erneut überschrieben wird. Der Wert, der mit OUT &EFFF, <Wert> gesendet wird, ist Bit-signifikant für die einzelnen Datenleitungen, dies bedeutet, das nullte Bit dieses Wertes entspricht der nullten Datenleitung, das erste Bit der ersten Datenleitung usw...

Ohne OUT nichts los

Durch das folgende Beispiel möchte ich dies veranschaulichen. Wenn ich nun also zum Beispiel den Befehl OUT &EFFF, &X00001001 eingebe, wird die Datenleitung 0 und 3 auf den Wert eins und die anderen auf den Wert null gesetzt. Am Centronics-Port macht sich dies nun dadurch bemerkbar, daß an den Pins 2 und 5 eine Spannung von +4 Volt gegenüber Masse anliegt. Über eine einfache Transistor-Verstärkerschaltung läßt sich mit dieser Schaltung ein Relais steuern (siehe Abbildung 1). Eine solche Verstärkerschaltung ist immer nötig, da die Ausgänge des Computers nur minimal belastet werden dürfen. Mit diesem Relais lassen sich nun die vielfältigsten Dinge steuern. Zwei einfache Fischertechnik-Modelle, die ich mit einem solchen Interface gesteuert habe, werde ich Ihnen gleich näher erklären und beschreiben. Es ist vielleicht ganz interessant, diese nachzubauen oder sie als Anreiz zu nehmen, selbst neue Modelle zu entwerfen.

An dieser Stelle ein wichtiger Hinweis: Ein Kurzschluß, das heißt, eine direkte Verbindung von einem Pin, der das Signal Eins führt, mit einem, der das Signal Null führt, kann Ihren Computer zerstören. Lassen Sie sich am besten durch einen Elektronik-Fachmann bei Ihren Experimenten beraten. So sollten Sie niemals direkt Kabel an den Centronics-Port halten oder gar löten, sondern immer über einen Stecker arbeiten, der im Elektronik-Fachgeschäft erhältlich ist. Die fertige Schaltung sollte in ein Gehäuse eingebaut werden, um sie so zu schützen. Als letzten Test können Sie die Schaltung an dem Stecker mit einem Widerstandsmeßgerät auf eventuelle Kurzschlüsse überprüfen, bevor Sie ihn in den Computer einstecken. Mit diesen Tips wollte ich Ihnen keine Angst einjagen, mein Computer lebt auch nach vielen Experimenten noch. Doch bitte ich Sie wirklich, diese Tips zu beherzigen.

Jetzt geht's ans Eingemachte

Doch nun zu einfachen Anwendungen dieser Schaltung. Entsprechend den sieben Datenleitungen des Centronics-Ports lassen sich natürlich auch bis zu sieben dieser Transistor-Verstärkerschaltungen nebeneinanderschalten. Je nach Anwendung läßt sich also ein Interface bauen, bei dem bis zu 7 Bit parallel vorliegen.

Um die Funktion dieses Interfaces zu beobachten, kann man natürlich ein-

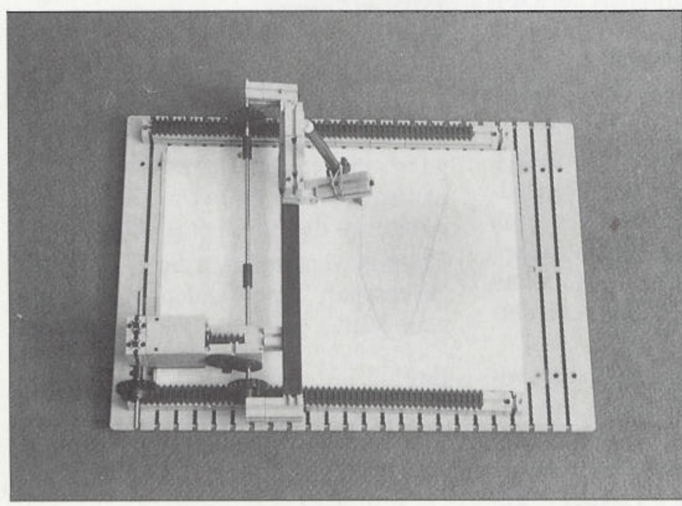


Bild 1: Der einfache Plotter

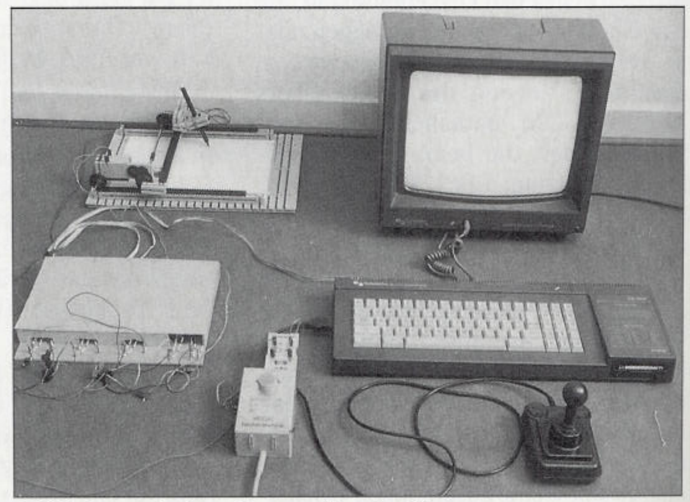


Bild 2: Der verdrahtete Gesamtaufbau des Plotters

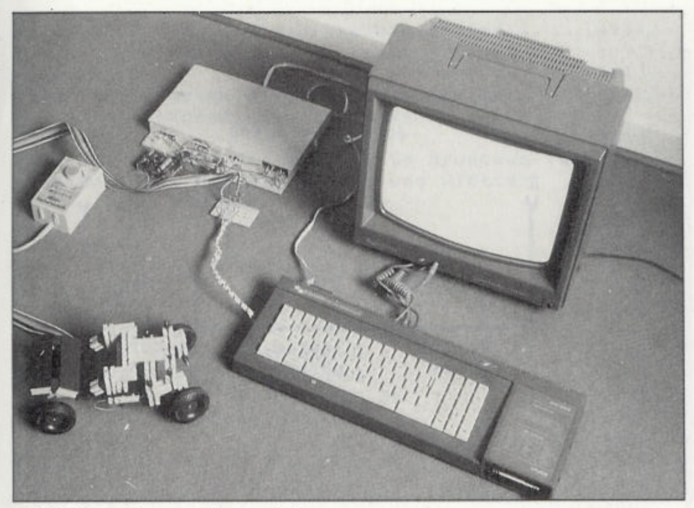


Bild 3: Der Gesamtaufbau mit Auto

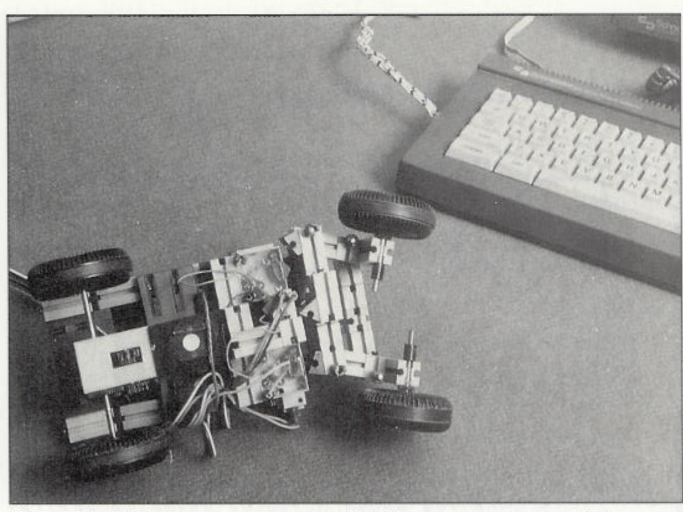


Bild 4: Das Fahrzeug von unten, die Positionstaster werden sichtbar

fach einmal ein Lämpchen mit dem Computer über das Relais ein- und ausschalten. Da ich mir kaum vorstellen kann, daß Sie sich damit zufriedengeben, möchte ich zwei einfache Modelle von mir vorstellen. Vor allen Dingen ist das zweite Modell, das Auto, recht interessant, da es einen einfachen Einstieg in die Robotersteuerung liefert. So ist es vom Benutzer aus frei programmierbar, doch dazu gleich mehr.

Was darf's sein: ein Auto oder ein Plotter?

Bei den Modellen handelt es sich um einen einfachen Plotter (Bild 1) und um ein programmierbares Auto. Für beide Modelle ist ein Interface mit nur 4 Bit nötig. Der Plotter sieht nun folgendermaßen aus. Es handelt sich um einen Stift der über zwei Motoren in X- und Y-Richtung beweglich ist. Dies läßt sich mit Fischertechnik am einfachsten mit Zahnstangen bewerkstelligen, auf denen ein Schlitten mit Zahnrädern läuft. Die beiden Motoren werden dem Schaltplan entsprechend mit dem Interface verdrahtet (siehe dazu Abbildung 2 und Bild 2). Zu diesem Aufbau habe ich nun zwei kleine Programme entworfen, um den Plotter zu steuern. Das erste besteht aus nur vier Zeilen. Mit diesem Programm kann man mit Joystick auf dem Plotter zeichnen. Durch die spezielle Verdrahtung des Plotters mit dem Interface ist es möglich, daß der Wert aus der Joystick-Abfrage direkt dem Wert entspricht, der zum Port gesendet werden muß.

Das zweite Programm dient dazu, mathematische Graphen zu zeichnen. Die Funktionszuordnung kann in Zeile 430 verändert werden. Die grobe Funktionsweise werde ich nun kurz darlegen. Das Programm arbeitet zeittaktgesteuert. Das bedeutet, daß der Plotter in 600x400 Punkte aufgeteilt ist und nun ein Zeittakt die Zeit ist, die der Plotter benötigt, um sich um eine Einheit fortzubewegen. So weiß der Computer bei bekanntem Ausgangspunkt auch ohne Rückmeldung, wo der Plotter im Moment steht, außerdem wird es durch die Zeittaktsteuerung möglich, genau definierte Strecken zu zeichnen, indem die Plottermotoren unterschiedlich lange angeschaltet werden (Zeile 470 bis 530). Da der Zeittakt plotterspezifisch ist, gibt es ein Unterprogramm "Taktwerte bestimmen" (Zeile 670 bis Ende). In diesem Unterprogramm wird die Gesamtzeit vom Computer gestoppt, die der Plotter in X- bzw. Y-Richtung benötigt, um das gesamte Blatt zu "durchfahren".

Diese Zeit wird dann durch 600 bzw. 400 geteilt, wodurch man die Taktwerte taktx bzw. takty erhält. Um nun den Graphen zu zeichnen, werden die Motoren feinstufig abwechselnd je nach Graph angestellt (Zeilen 410 bis 570). Gleichzeitig wird der Graph auch auf dem Bildschirm dargestellt. Die Werte maxx bzw. maxy legen die Größe des Koordinatensystems fest. Der Punkt maxx, maxy liegt also sowohl auf dem Plotter als auch auf dem Bildschirm rechts oben im Koordinatensystem.

Nun zu dem zweiten Modell, dem Auto (Bild 3). Zuerst der Aufbau: Es ist wiederum nur ein 4-Bit-Interface nötig. Es werden zwei Motoren benötigt. Einer als Antrieb, ein zweiter zur Lenkung.

Achtung, jetzt kommt ein Auto

Die Lenkung wird entweder als Zahnstangenlenkung oder als Lenkung, bei der die gesamte vordere Achse um einen Punkt drehbar ist, realisiert. Wichtig für die Lenkung sind drei einfache Taster. Diese werden so montiert, daß ein Taster bei vollem Lenkausschlag links, der andere bei vollem Lenkausschlag rechts und der dritte bei Geradeausfahrt gedrückt wird. Die Motoren und die Taster werden dem Schaltplan entsprechend verdrahtet (der gleiche

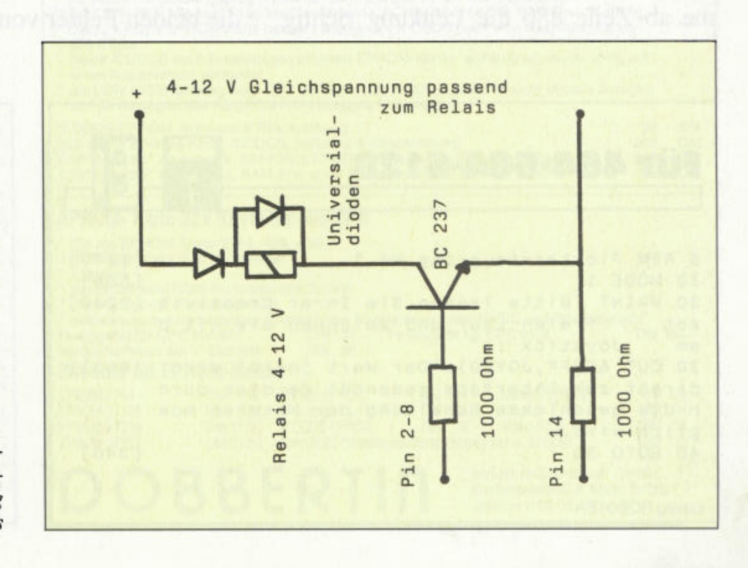


Abbildung 1: Schaltbild der Transistor-Verstärkerschaltung mit Relais

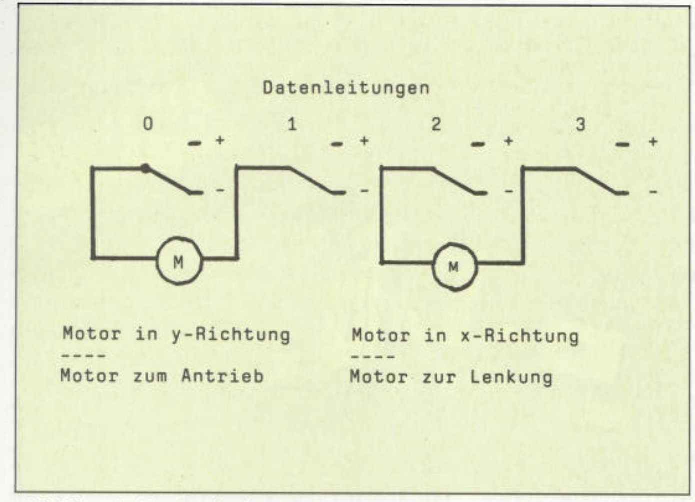


Abbildung 2: Die Verdrahtung der Plottermotoren

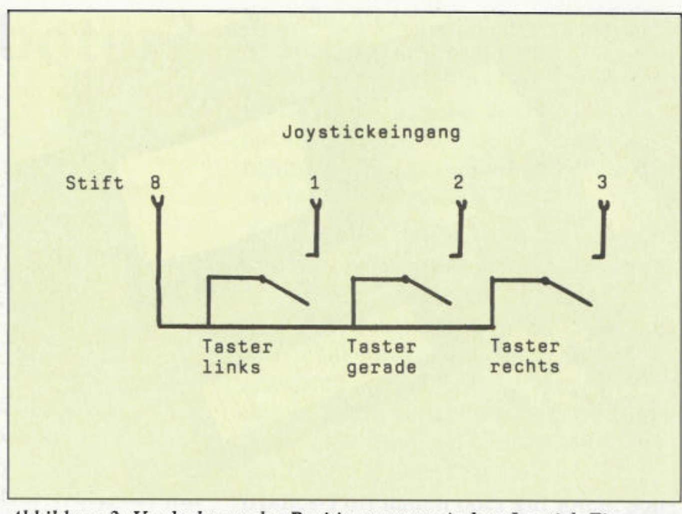


Abbildung 3: Verdrahtung der Positionstaster mit dem Joystick-Eingang

Schaltplan wie beim Plotter, wobei der Antriebsmotor dem Plottermotor in Y-Richtung und der Lenkungsmotor dem Plottermotor in X-Richtung entspricht). Die Taster werden an den Joystick-Eingang angeschlossen (siehe Abbildung 3), dazu besorgt man sich am besten aus dem Elektronik-Fachgeschäft einen entsprechenden Stecker.

Beim Bau des Autos sind der Phantasie keine Grenzen gesetzt, nur bei der Verdrahtung der Motoren und der Taster sollte man sich an den Schaltplan halten, da man sonst das Programm nicht mehr anwenden kann. Nun zum Programm, mit dem sich das Auto steuern läßt: Das Programm hat drei Teilprogramme.

1. Manuelles Steuern mit Cursortasten
2. Programmieren eines Fahrtablaufs
3. Ablauf eines Programms

Zuerst zum ersten Teilprogramm. Dies hat folgenden Ablauf: Es wird abgefragt, wohin der Bediener fahren will, dann wird mit Hilfe der Unterprogramme ab Zeile 880 die Lenkung richtig eingestellt und dann wird der Fahrtmotor eingeschaltet.

Nun zum zweiten Teilprogramm "Programmablauf einprogrammieren": Die Speicherung der Programme erfolgt in zwei Feldern, die direkt in Zeile 10 angelegt werden. Der Wert der indizierten Variable "aktion" kennzeichnet, was das Auto machen soll, so sind den einzelnen Aktionen geradeaus, rückwärts, links usw. Werte von 0 bis 5 zugeordnet. Der Wert der indizierten Variable "Länge" gibt an, wie lange die jeweilige Aktion durchgeführt werden soll. Die erste Indexzahl kennzeichnet jeweils die Nummer des Programms, die zweite die Nummer des jeweiligen Programmschritts. Der Ablauf sieht nun so aus, daß zuerst registriert wird, welche Taste gedrückt ist (Zeilen 480 bis 550), dann wird die gewünschte Aktion abgespeichert (560 bis 570), und danach wird die gewünschte Aktion ausgeführt (580 bis 620).

Im letzten Teilprogramm, Abruf eines Programmablaufs, werden nun einfach die beiden Felder von Anfang an abgearbeitet und die jeweilige Aktion mit der gewünschten Länge ausgeführt, bis zum Stop-Zeichen (aktion=5). Ich hoffe, der Artikel hat Sie ermuntert, selbst etwas zu versuchen und über diese beiden Modelle den Einstieg zu finden.

Es kann auch bestellt werden

Da vielleicht der ein oder andere keine Zeit dafür aufbringen kann, sich solch ein Interface herzustellen, möchte ich hier anbieten, dies für Sie zu günstigen Preisen zu übernehmen. Das Interface wird etwas kompakter als auf dem Foto ausfallen.

Der Preis für ein 7-Bit-Interface beträgt 150,– DM.

Wenn Sie ein solches Interface bestellen möchten oder wenn Sie sonst noch hier und da Fragen haben, wenden Sie sich vertrauensvoll an mich.

Klaus Mayer,
Höhenweg 73,
5630 Remscheid 11,
Tel. 02191/67785

(cd)

für 464-664-6128

```
5 REM Plottersteuerung Nr.1                  [1277]
10 MODE 1                                    [506]
20 PRINT "Bitte lassen Sie Ihrer Creativit [9040]
aet      freien Lauf,und zeichnen sie mit d
em    Joystick !!"
30 OUT &EFFF,JOY(0):'Der Wert joy(0) wird  [8531]
direkt zum Interface gesendet da dies durc
h die geschickte Schaltung der Motoren moe
glich wird.
40 GOTO 30                                   [340]
```

Listing ROBOTER

```
1 REM Plottersteuerung Nr.2                  [1276]
10 ' ***** Vorbereitung *****                [1127]
20 MODE 1                                    [506]
30 PRINT "Kennen Sie Ihre plotterspezifisc [6123]
hen     Taktwerte (j/n))"
40 WHILE ein$<>"j" AND ein$<>"n"             [1403]
50 ein$=INKEY$                               [545]
60 WEND                                      [390]
70 IF ein$="j" THEN GOSUB 610 ELSE GOSUB 6 [2013]
70
80 MODE 1                                    [506]
90 PRINT "Fahren Sie bitte in die linke ob [5792]
ere Eckeund druecken Sie <Space>"
100 WHILE INKEY$<>" "                        [1292]
110 OUT &EFFF,JOY(0)                         [80]
120 WEND                                     [390]
130 MODE 1                                   [506]
```

Listing ROBOTER

```
140 INPUT "Geben Sie bitte die maximale      [4904]
     X-Koordinate an";maxx
150 PRINT                                    [361]
160 INPUT "Geben Sie bitte die maximale      [4043]
     Y-Koordinate an";maxy
170 PRINT                                    [361]
180 PRINT "Wenn Sie eine Taste druecken ,e [8993]
rhalten  Sie auf Bildschirm und Plotter
        den Graphen"
190 IF INKEY$="" THEN 190                    [762]
200 '****                                    [104]
210 '**** Koordinatensystem *****            [1223]
220 '****                                    [104]
230 MODE 1                                   [506]
240 ORIGIN 320,200                           [215]
250 MOVE -300,0:DRAW 300,0                   [1231]
260 MOVE 0,-200:DRAW 0,200                   [946]
270 FOR x=-300 TO 300 STEP 30                [1791]
280 MOVE x,-5:DRAW x,5                       [1467]
290 NEXT                                     [350]
300 FOR y=-180 TO 180 STEP 30                [1939]
310 MOVE -5,y:DRAW 5,y                       [2071]
320 NEXT                                     [350]
330 PRINT "Maxx:";maxx                       [1679]
340 PRINT "Maxy:";maxy                       [1806]
350 MOVE -300,0                              [874]
360 xposi=-300:' ** Dem Computer wird die [4766]
Position des Plotters,oben links,mitgeteil
t **
370 yposi=200                                [1121]
380 '***                                     [53]
390 '*** Gleichzeitig auf Bildschirm und P [3915]
lotter zeichnen ***
400 '***                                     [53]
410 FOR x=-maxx TO maxx STEP maxx/40         [2956]
420 neuexpos=x*300/maxx                      [2174]
430 y=SIN(x)                                 [972]
440 neueypos=y*200/maxy                      [2261]
450 DRAW neuexpos,neueypos                   [1270]
```

Listing ROBOTER

```
460 IF neuexpos>xposi THEN OUT &EFFF,&X100 [2250]
0 ELSE OUT &EFFF,&X100
470 zeit=TIME                                [990]
480 WHILE ABS(zeit-TIME)<ABS((neuexpos-xpo [2219]
si)*taktx):WEND
490 OUT &EFFF,&X0                            [517]
500 IF neueypos>yposi THEN OUT &EFFF,&X1 E [1919]
LSE OUT &EFFF,&X10
510 zeit=TIME                                [990]
520 WHILE ABS(zeit-TIME)<ABS((neueypos-ypo [3949]
si)*takty):WEND
530 OUT &EFFF,&X0                            [517]
540 xposi=neuexpos                           [1538]
550 yposi=neueypos                           [1130]
560 NEXT                                     [350]
570 END                                      [110]
580 '***                                     [53]
590 '*** Unterprogramm Taktwerte bekannt * [3330]
**
600 '***                                     [53]
610 MODE 1                                   [506]
620 INPUT "Bitte geben Sie ihre plotterspe [9700]
zifischenTaktwerte durch Komma getrennt ei
n:      ";taktx,takty
630 RETURN                                   [555]
640 '***                                     [53]
650 '*** Unterprogramm Taktwerte bestimmen [3169]
 ***
660 '***                                     [53]
670 MODE 1                                   [506]
680 PRINT "Fahren Sie bitte in die linke o [5185]
bere Eckeund druecken Sie dann <Space>"
690 WHILE INKEY$<>""                         [1292]
700 OUT &EFFF,JOY(0)                         [80]
710 WEND                                     [390]
720 PRINT                                    [361]
730 PRINT "Sobald Sie <Space> druecken,fae [12597]
hrt der  Plotter nach rechts,wenn er am re
chten  Rand angelangt ist,druecken Sie bit
```

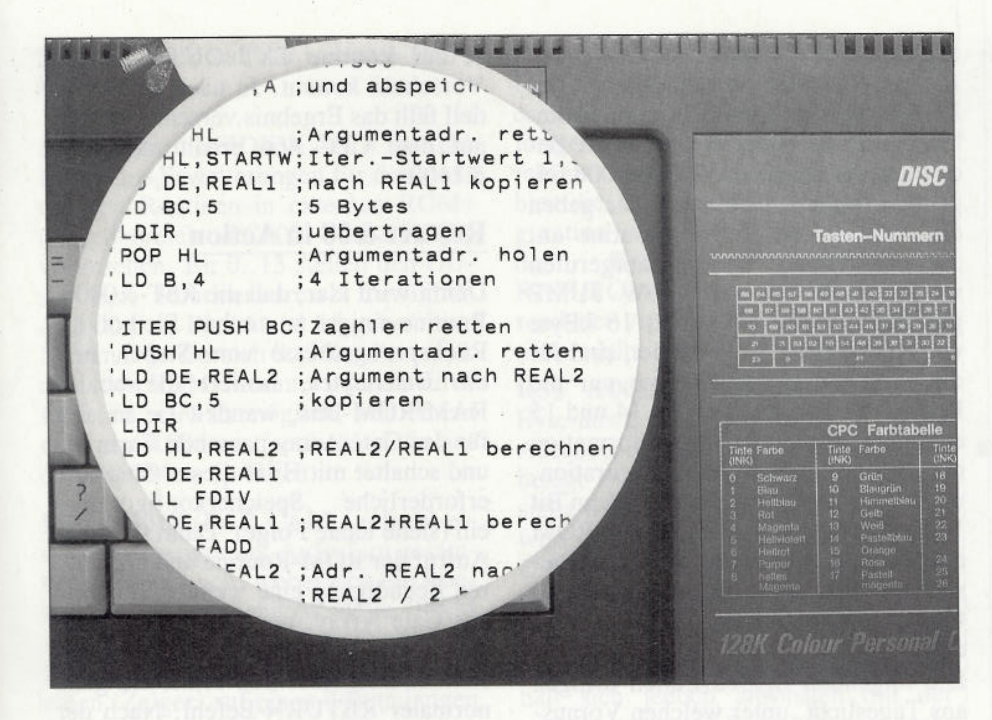
Listing ROBOTER

```
te   sofort <Space> um ihn zu stoppen"
740 IF INKEY$<>" " THEN 740              [775]
750 zeit=TIME                             [990]
760 WHILE INKEY$<>" "                     [1292]
770 OUT &EFFF,&X1000                      [754]
780 WEND                                  [390]
790 OUT &EFFF,&X0                         [517]
800 taktx=(TIME-zeit)/600                 [1946]
810 PRINT                                 [361]
820 PRINT "Sobald Sie <Space> druecken,fae [13597]
hrt der  Plotter nach unten,wenn er am unt
eren   Rand angelangt ist,druecken Sie bit
te   sofort <Space> um ihn zu stoppen"
830 IF INKEY$<>" " THEN 830              [781]
840 zeit=TIME                             [990]
850 WHILE INKEY$<>" "                     [1292]
860 OUT &EFFF,&X10                        [672]
870 WEND                                  [390]
880 OUT &EFFF,&X0                         [517]
890 takty=(TIME-zeit)/400                 [1742]
900 MODE 1                                [506]
910 PRINT                                 [361]
920 PRINT "Ihre druckerspezifischen Taktwe [5270]
rte      lauten: ";taktx;",";takty
930 PRINT                                 [361]
940 PRINT "Um fortzufahren druecken Sie bi [5314]
tte <Space>."
950 IF INKEY$<>" " THEN 950              [765]
960 RETURN                                [555]
```

```
5 REM Autosteuerung                        [824]
10 DIM aktion(30,100),laenge(30,100)       [2739]
20  '**************                        [345]
30  '**** Menue ****                       [113]
40  '**************                        [345]
50 MODE 1                                  [506]
60 PRINT "Mit diesem Programm koennen Sie  [19478]
Ihren    Wagen sowohl einfach manuell mit d
en   Cursortasten steuern,als auch einen
     Fahrtablauf einprogrammieren und
   diesen dann beliebig oft wiederholen
lassen.Es stehen Ihnen bis zu 100
70 PRINT "Programme zur Verfuegung.Viel Sp [4755]
ass!!"
80 PRINT                                   [361]

90 PRINT                                   [361]
100 PRINT "    1. Manuelles Steuern"       [2809]
110 PRINT "    2. Einprogrammieren eines   [5239]
           Fahrtablaufs"
120 PRINT "    3. Ablauf eines Programms"  [2905]
130 PRINT "    4. Ende"                    [2060]
140 PRINT                                  [361]
150 INPUT " Bitte waehlen Sie";wahl        [2367]
160 IF wahl=4 THEN 190                     [1001]
170 ON wahl GOSUB 230,360,700              [893]
180 GOTO 50                                [384]
190 END                                    [110]
200  '*************************            [978]
210  '** Steuern mit Joystick **           [1555]
220  '*************************            [978]
230 MODE 1                                 [506]
240 PRINT "Mit den Cursortasten koennen Si [5573]
e Ihr    Auto nun beliebig steuern."
250 taste=0                                [851]
260 IF INKEY(0)<>-1 THEN GOSUB 1090:OUT &E [4119]
FFF,1:taste=1
270 IF INKEY(2)<>-1 THEN GOSUB 1090:OUT &E [2427]
FFF,2:taste=2
280 IF INKEY(8)<>-1 THEN GOSUB 910:OUT &EF [3112]
FF,1:taste=3
290 IF INKEY(1)<>-1 THEN GOSUB 1000:OUT &E [3560]
FFF,1:taste=4
300 IF taste=0 THEN OUT &EFFF,0            [1533]
310 IF INKEY(60)<>-1 THEN 320 ELSE 250     [1342]
320 RETURN                                 [555]
330  '**********************************   [1460]
340  '* Programmablauf einprogrammieren *  [2810]
350  '**********************************   [1460]
360 MODE 1                                 [506]
370 PRINT                                  [361]
380 INPUT " Unter welcher Nummer soll der  [5288]
         Programmablauf abgespeichert wer
den";nr
390 PRINT                                  [361]
400 PRINT " Sobald Sie eine Taste druecken [13525]
,koennen  Sie mit den Cursortasten den Wag
en   steuern,der Fahrtablauf wird
     automatisch unter der angegebenen
   Nummer einprogrammiert."
410 IF INKEY$="" THEN 410                  [886]
420 MODE 1                                 [506]
430 PRINT " Bitte steuern Sie nun den Wage [5926]
n nach   Ihren Wuenschen!!"
440 schritt=1                              [704]
450 taste=0                                [851]
460 zeit=TIME                              [990]
470 alttaste=taste                         [542]
480 WHILE taste=alttaste                   [1353]
490 taste=0                                [851]
500 IF INKEY(0)<>-1 THEN taste=1           [2181]
510 IF INKEY(2)<>-1 THEN taste=2           [1526]
520 IF INKEY(8)<>-1 THEN taste=3           [2011]
530 IF INKEY(1)<>-1 THEN taste=4           [1999]
540 IF INKEY(60)<>-1 THEN OUT &EFFF,0:tast [2002]
e=5
550 WEND                                   [390]
560 aktion(nr,schritt)=alttaste            [2212]
570 laenge(nr,schritt)=TIME-zeit           [1717]
580 IF taste=0 THEN OUT &EFFF,0            [1533]
590 IF taste=1 THEN GOSUB 1090:OUT &EFFF,1 [2949]
600 IF taste=2 THEN GOSUB 1090:OUT &EFFF,2 [1860]
610 IF taste=3 THEN GOSUB 910:OUT &EFFF,1  [2031]
620 IF taste=4 THEN GOSUB 1000:OUT &EFFF,1 [2754]
630 schritt=schritt+1                      [1204]
640 IF taste<>5 THEN 460                   [1835]
650 aktion(nr,schritt)=5                   [1520]
660 RETURN                                 [555]
670  '******************************       [1223]
680  '* Abruf eines Programmablaufs *      [974]
690  '******************************       [1223]
700 MODE 1                                 [506]
710 INPUT " Geben Sie bitte die Nummer des [8284]
      Programms ein,das Sie ablaufen l
assen    wollen.";nr
720 PRINT                                  [361]
730 PRINT " Sobald Sie eine Taste druecken [5704]
,laeft     das Programm ab und steuert Ihre
n Wagen"
740 IF INKEY$="" THEN 740                  [908]
750 schritt=1                              [704]
760 WHILE aktion(nr,schritt)<>5            [2467]
770 IF aktion(nr,schritt)=1 THEN GOSUB 109 [2700]
0:OUT &EFFF,1
780 IF aktion(nr,schritt)=2 THEN GOSUB 109 [3168]
0:OUT &EFFF,2
790 IF aktion(nr,schritt)=3 THEN GOSUB 910 [3763]
:OUT &EFFF,1
800 IF aktion(nr,schritt)=4 THEN GOSUB 100 [3359]
0:OUT &EFFF,1
810 IF aktion(nr,schritt)=0 THEN OUT &EFFF [1510]
,0
820 zeit=TIME                              [990]
830 WHILE TIME-zeit<laenge(nr,schritt):WEN [2377]
D
840 schritt=schritt+1                      [1204]
850 WEND                                   [390]
860 OUT &EFFF,0                            [271]
870 RETURN                                 [555]
880  '*****************                    [665]
890  '*** links lenken ***                 [1296]
900  '*****************                    [665]
910 IF JOY(0)=1 THEN 960                   [574]
920 OUT &EFFF,8                            [263]
930 WHILE JOY(0)<>1                        [1043]
940 WEND                                   [390]
950 OUT &EFFF,0                            [271]
960 RETURN                                 [555]
970  '******************                   [657]
980  '*** rechts lenken ***                [806]
990  '******************                   [657]
1000 IF JOY(0)=4 THEN 1050                 [1169]
1010 OUT &EFFF,4                           [267]
1020 WHILE JOY(0)<>4                       [1052]
1030 WEND                                  [390]
1040 OUT &EFFF,0                           [271]
1050 RETURN                                [555]
1060  '*********************               [768]
1070  '*** geradeaus lenken ***            [2848]
1080  '*********************               [768]
1090 IF JOY(0)=2 THEN 1120                 [988]
1100 IF JOY(0)=1 THEN OUT &EFFF,4:WHILE JO [2995]
Y(0)<>2:WEND:OUT &EFFF,0
1110 IF JOY(0)=4 THEN OUT &EFFF,8:WHILE JO [3522]
Y(0)<>2:WEND:OUT &EFFF,0
1120 RETURN                                [555]
```

Listing ROBOTER



Die Assemblerecke

Bankgeheimnisse

Wenn der verfügbare Speicher den Adreßbereich des Prozessors übersteigt, so muß sich eine ausgefeilte Schaltlogik um das Ein- und Ausblenden überzähliger Speicherbänke kümmern. Das 'Banking' wird oft verwendet, um die Leistungsfähigkeit von 8-Bit-Systemen zu erhöhen. Bei modernen 16-Bit-Rechnern sollte diese umständliche Technik eigentlich der Vergangenheit angehören – doch das ist "ein typischer Fall von denkste"! In PC-Kreisen hört man allerorten Klagen über die beschränkten 640 kByte Arbeitsspeicher unter MS-DOS, und die Anwender schlagen sich mit einer Erweiterung namens EMS herum, die zusätzliche RAM-Bänke für Desktop-Publisher und andere unersättliche Software einblendet... Grund genug, um zu erforschen, wie das nach wie vor aktuelle Banking im CPC funktioniert!

Bereits in der letzten Folge der Assemblerecke wurde die Umschaltung zwischen RAM- und ROM-Speicherbänken über das Gate Array erörtert. In den internen Registern dieses Bausteins befinden sich bestimmte Bits, die die 'Hardware-Schalter' für das Banking darstellen. Es ist jedoch noch eine Menge zusätzliche Verwaltungsarbeit notwendig, damit Anwender und Programmierer unbeschwert auf ihrem CPC herumtippen können, ohne sich ständig um die Speicherkonfiguration kümmern zu müssen. Diese Arbeit leistet das Betriebssystem und benutzt dabei spezielle Maschinenbefehle, mit denen sich eine Art Befehlserweiterung für den Z80-Prozessor realisieren läßt.

Die sogenannten 'Restart-Befehle' (Assembler-Kürzel: RST) sind eigentlich nichts anderes als besonders rationell organisierte Unterprogramm-Aufrufe. Sie funktionieren im Prinzip genau wie CALL-Befehle, benötigen jedoch keine Zieladresse, da diese bereits im Prozessor eingebaut ist. Deshalb belegen RST-Kommandos nur ein Byte im Programmcode und werden sehr schnell ausgeführt. Allerdings existieren auch nur acht fest verdrahtete Adressen, die sich auf diese Weise erreichen lassen. Im Vergleich zu CALL wurde hier also die Geschwindigkeit auf Kosten der Flexibilität erhöht.

Kellerkinder

Die acht RST-Zieladressen liegen allesamt im 'Keller' des CPC-Speichers: Die erste bei &0000, die zweite bei &0008, und so weiter, in Schritten zu jeweils 8 Byte. In einem Assemblerprogramm wird die Adresse als Parameter an das RST-Kommando angehängt (z.B. RST &0008), taucht aber wie bereits gesagt nicht explizit im Maschinencode auf, sondern geht in den 1-Byte-Opcode des Befehls ein (implizite Adressierung). In der Literatur werden die Restart-Befehle mitunter von 0 bis 7 durchnumeriert; die Angabe RST 5 entspricht dann der Zieladresse 5 x 8 = 40. Manche Assembler verstehen auch diese Schreibweise. Der in Heft 6/88 abgedruckte CPC-Assembler besteht jedoch auf der Adreßangabe und weist ungültige Werte mit einer Fehlermeldung zurück.

Zwischen den Restart-Adressen stehen also genau 8 Byte für eine kurze Routine zur Verfügung. Will man ein längeres Unterprogramm aufrufen, so muß man an die entsprechende RST-Adresse einen JP-Befehl setzen, der zu der eigentlichen Routine springt. Diese kann dann irgendwo im Speicher stehen und natürlich beliebig lang sein. Allerdings sind bis auf RST &0030 alle Restart-Befehle im CPC mit wichtigen Betriebssystem-Funktionen vorbelegt, so daß eine Umdefinition für eigene Zwecke kaum möglich ist. Die Tabelle informiert über den Sinn und Zweck der einzelnen Einsprünge und zeigt, daß insgesamt fünf Restarts mit dem RAM/ROM-Status zu tun haben.

Einige der Routinen (z.B. FAR CALL) lesen die Rückkehradresse auf dem Stack, um auf Datenbytes zugreifen zu können, die auf den RST-Befehl folgen und eine Adresse oder Angaben über die gewünschte Speicherkonfiguration enthalten. In einem Assemblerprogramm wird diese Zusatzinformation mit Hilfe der Direktiven DB oder DW eingefügt. Natürlich erhöht die RST-Routine die Rückkehradresse um 2, damit der Programmablauf beim Rücksprung nicht direkt nach dem RST-Befehl, sondern erst nach den Datenbytes fortgesetzt wird. Beim LOW JUMP und FIRM JUMP wird die Rückkehradresse sogar ganz vom Stack entfernt, so daß sich diese Aufrufe nicht mehr wie ein CALL, sondern wie ein normaler Sprungbefehl (JP) verhalten.

Wir werden diese trickreiche Verwendung der RST-Befehle gleich anhand von Beispielen durchleuchten. Zunächst aber ein paar Betrachtungen über den RAM/ROM-Status des CPC in verschiedenen Situationen: Während der Ausführung eines BASIC-Programms muß das obere ROM aktiv sein (Adreßbereich &C000 bis &FFFF), da es den Interpreter-Maschinencode enthält. Das BASIC-Programm steht natürlich im RAM; das Betriebssy-

stem-ROM im unteren Adreßbereich (&0000 bis &3FFF) ist zu diesem Zeitpunkt abgeschaltet. Wenn Sie von BASIC aus ein Maschinenprogramm starten, wird auch das obere ROM ausgeblendet, um einen direkten Zugriff auf das parallel liegende Video-RAM zu ermöglichen.

Der lange Weg ins ROM

Angenommen, der Interpreter oder Ihr Maschinenprogramm ruft nun die Betriebssystem-Routine TXT OUTPUT mit CALL &BB5A auf, um ein Zeichen auf dem Bildschirm auszugeben. Offenbar sind jetzt drei Schritte notwendig:

- Einschalten des Betriebssystem-ROM (damit die Routine ausgeführt werden kann) und Ausschalten des oberen ROM (es muß ja etwas in den Bildschirmspeicher geschrieben werden).
- Abarbeiten von TXT OUTPUT
- Rücksprung und Wiederherstellen des vorherigen RAM/ROM-Status.

Um diesen Vorgängen auf die Spur zu kommen, werfen wir einen kurzen Blick in die Speicherstelle &BB5A. PRINT HEX$(PEEK(&BB5A)) bringt das Byte &CF zum Vorschein, und das ist genau der Opcode für den Befehl RST &0008 (LOW JUMP). Analysiert man den Maschinencode ab Adresse &0008, so findet man beim CPC 464 den Befehl JP &B982 (CPC 664/6128: JP &B98A), der zu der eigentlichen Restart-Routine springt. Um zu verstehen, wie sie arbeitet, muß man die Bedeutung der beiden auf RST &0008 folgenden Datenbytes kennen. Sie geben die Adresse der (ROM-)Routine an, die über diesen Umweg aufgerufen werden soll. Da ein LOW JUMP grundsätzlich in den ersten 16 kByte von &0000 bis &3FFF landet, sind für die Darstellung der Adresse nur die Bits 0...13 erforderlich. Bit 14 und 15 enthalten als zusätzliche Information die gewünschte Speicherkonfiguration, und zwar in folgender Form: Wenn Bit 14 eine 0 enthält, wird das untere ROM eingeblendet, ansonsten bleibt es abgeschaltet. Bit 15 ist auf gleiche Weise für das obere (BASIC-)ROM zuständig.

Die folgenden BASIC-Zeilen bringen ans Tageslicht, unter welchen Voraussetzungen der Aufruf von TXT OUTPUT über Adresse &BB5A abläuft:

```
10 adr=PEEK(&BB5B)+256*PEEK(&BB5C)
20 adr=UNT(adr)
30 PRINT BIN$(adr,16)
40 PRINT HEX$(adr AND &3FFF,4)
```

Zeile 30 gibt den Inhalt der Datenbytes binär aus, so daß Sie die Bits 14 und 15 in Augenschein nehmen können. Wie zu erwarten, findet man in Bit 14 eine 0 (unteres ROM ein) und in Bit 15 eine 1 (oberes ROM aus bzw. Video-RAM ein). Zeile 40 eliminiert die beiden Speicher-Bits, so daß die ROM-Adresse der Routine TXT OUTPUT zum Vorschein kommt. Je nach CPC-Modell fällt das Ergebnis verschieden aus; auf dem CPC 464 erhält man z.B. &1400.

Restart &08 in Action

Damit wird klar, daß die RST-&0008-Routine einiges zu tun hat. Sie holt die Rücksprungadresse vom Stapel, liest die Datenbytes, isoliert die beiden RAM/ROM-Bits, wandelt sie in eine für das Gate Array passende Form um und schaltet mit Hilfe dieses Chips die erforderliche Speicherkonfiguration ein (siehe letzte Folge). Dann folgt der Aufruf der ROM-Routine und nach deren Beendigung eine weitere Meldung ans Gate Array, um den vorher gültigen RAM/ROM-Status wieder herzustellen. Den Abschluß bildet ein ganz normaler RETURN-Befehl. Nach der verwirrenden Herumhopserei im CPC-Speicher stellt sich allerdings die berechtigte Frage, wo der Programmablauf jetzt fortgesetzt wird. Die zu dem RST-Befehl gehörige Rückkehradresse wurde ja vom Stapel entfernt, und was liegt darunter? Natürlich die durch CALL &BB5A erzeugte Rückkehradresse, so daß der Z80 ganz brav mit dem nächsten Befehl nach dem Aufruf von TXT OUTPUT weitermacht.

Alle Betriebssytemaufrufe im CPC werden über diesen Mechanismus abgewickelt. Ab Adresse &BB00 findet man eine ganze Serie von RST-&0008-Befehlen mitsamt den dazugehörigen Datenbytes. Sie werden nach dem Einschalten des CPC aus dem Betriebssystem-ROM in den zentralen RAM-Bereich kopiert, wo sie unabhängig von der Speicherkonfiguration jederzeit erreichbar sind.

Seitensprünge und Ferngespräche

Diese 'Sprungleiste' organisiert nicht nur die Umschaltung des RAM/ROM-Status, sondern sorgt auch dafür, daß die Einsprungstellen ins Betriebssystem immer an der gleichen Stelle liegen, obwohl die ROM-Adressen der Routinen je nach CPC-Version unterschiedlich sind. Ein unübersehbarer Nachteil ist jedoch der Zeitverlust, der bei jedem Aufruf durch die Abarbeitung der RST &0008-Routine entsteht. Dieses Problem läßt sich in Banking-Systemen kaum vermeiden; das Ausbügeln von Hardware-Beschränkungen durch Software-Tricks fordert letztendlich doch seinen Tribut.

Die RST-Routinen im CPC

Einsprung	Name	Bedeutung
RST &0000	RESET ENTRY	Zurücksetzen des Rechners (wie CALL 0)
RST &0008	LOW JUMP*	Sprung auf eine Routine in den unteren 16 KByte (&0000-&3FFF) mit Angabe der RAM/ROM-Konfiguration.
RST &0010	SIDE CALL*	Aufruf einer Routine in einem Erweiterungs-ROM (&C000-&FFFF) bei externen Hardware-Erweiterungen.
RST &0018	FAR CALL*	Aufruf einer beliebigen Routine im gesamten Adressbereich mit Angabe der RAM/ROM-Konfiguration.
RST &0020	RAM LAM	Lesen eines Bytes aus dem RAM unabhängig vom RAM/ROM-Status. Entspricht LD A,(HL).
RST &0028	FIRM JUMP*	Sprung auf eine beliebige Routine im gesamten Adressbereich bei gleichzeitiger Aktivierung des unteren ROMs.
RST &0030	USER	Frei verwendbar
RST &0038	INTERRUPT ENTRY	Einsprung für den CPC-Systeminterrupt

* Diese Routinen erwarten, daß dem RST-Befehl 2 Datenbytes folgen

Die Restart-Befehle, kurz RST genannt, entsprechen im Prinzip den CALL-Befehlen

Nun ein Blick auf die weiteren RST-Befehle:

RST &0010 (SIDE CALL) ist bei Hardware-Erweiterungen für den Aufruf von Routinen in externen ROMs (Adreßbereich &C000 bis &FFFF) vorgesehen. Bit 0..13 stellen den Offset der Routinenadresse bezüglich &C000 dar; Bit 14 und 15 geben eine Nummer im Bereich 0..3 an, die zu der aktuellen ROM-Nummer addiert wird. Dadurch wird es möglich, von einem externen ROM aus 'seitlich' in ein anderes ROM zu springen.

RST &0018 (FAR CALL) erlaubt einen Unterprogramm-Aufruf im gesamten Adreßbereich, wobei sich die Routine an jeder beliebigen Stelle (RAM oder ROM) befinden darf. Die beiden Datenbytes hinter dem Befehl stellen einen 'Zeiger' auf einen 3 Byte langen Parameterblock dar, der einige zusätzliche Informationen für die RST-&0018-Routine zur Verfügung stellen muß: In den ersten beiden Bytes die Aufrufadresse der Routine, im dritten Byte die erforderliche Speicherkonfiguration. Werte von &00 bis &FB aktivieren im oberen Adreßbereich (&C000 bis &FFFF) ein (externes) ROM mit der entsprechenden Nummer. Das BASIC-ROM trägt zum Beispiel die Nummer 0, das Floppy-ROM die Nummer 7. Die Werte &FC..FF haben einen Sonderstatus: Bit 0 und Bit 1 bestimmen hier, ähnlich wie beim LOW JUMP, den ROM/RAM-Status. Bit 0 ist für den unteren, Bit 1 für den oberen Adreßbereich zuständig. Steht das Bit auf 0, so ist dort ROM selektiert, ansonsten RAM.

Mit dem FAR CALL kommt man also an jede Routine im CPC heran, wo immer sie sich auch verstecken mag. Hier ein Beispiel zur Illustration: Angenommen, Sie möchten die Routine für die vorzeichenlose Multiplikation zweier Integer-Zahlen benutzen. Sie befindet sich beim CPC 464 im unteren ROM und ist bequem über die Einsprungstelle &BD5B zu erreichen. Im CPC 664/6128 wurde die Integer-Arithmetik ins obere BASIC-ROM verlegt (Adresse &DD60 bzw. &DD5B) und der Einsprungvektor eliminiert. Also basteln wir uns selbst einen und versehen ihn mit dem Label INTMUL:

```
100 ';*** Multiplikation HL=HL*DE
110 'INTMUL RST &18 ;Far Call
120 'DW PBLOCK ;Adr. Parameterblock
130 'RET
140 ';*** Far Call-Parameterblock
150 'PBLOCK DW &DD5B ;6128-Adresse
160 'DB &FD ;oberes ROM ein
```

Zu beachten ist, daß die Abarbeitung nach dem Rücksprung wie bei einem echten CALL hinter den Datenbytes fortgesetzt wird; deshalb der RET-Befehl in Zeile 130. Natürlich wird auch hier die ursprüngliche Speicherkonfiguration wieder hergestellt; der mit dem Byte &FD spezifizierte RAM/ROM-Status gilt also nur während der Ausführung der Multiplikationsroutine.

RST &0020 (RAM LAM) liest ein Byte aus dem RAM, selbst wenn der Adreßbereich gerade von einem ROM belegt ist. Die Adresse der Speicherstelle muß in HL stehen, das Ergebnis wird im A-Register zurückgeliefert. Dieser Restart verhält sich also genau wie der Befehl LD A,(HL).

RST &0028 (FIRM JUMP) schaltet das untere (Firmware-)ROM ein, verhält sich jedoch ansonsten mit der nachfolgenden 2-Byte-Adresse exakt wie ein JP-Befehl, da die Rückkehradresse vom Stack entfernt wird. Die neue Speicherkonfiguration ist gültig bis zur Beendigung der angesprungenen Routine mit RET. Der FIRM JUMP wird im CPC für die Arithmetik-Einsprünge benutzt (siehe Heft 8/89). Die Routinen befinden sich im unteren ROM. Da sie nicht auf den Bildschirmspeicher zugreifen, kann auf ein Ausblenden des oberen ROM und damit auf den etwas komplizierteren LOW JUMP verzichtet werden.

RST &0030 (USER) steht Ihnen für eigene Ideen zur Verfügung. Wenn in Ihrem Programm ein Unterprogramm besonders häufig benutzt wird, so können Sie den Aufruf über den RST &0030 laufen lassen, indem Sie an dieser Adresse einen Sprungbefehl auf die Routine eintragen. Im Vergleich zu CALL sparen Sie 2 Bytes pro Aufruf.

RST &0038 (INTERRUPT ENTRY) wird von der Hardware 300mal pro Sekunde ausgelöst und stellt den Einsprung in die zentrale CPC-Interruptroutine dar. Wer hier dran dreht, muß genau wissen, was er tut, sonst ist ein Crash unvermeidlich!

Damit wären einige der "Bankgeheimnisse" des CPC gelüftet. Wer raffinierte Programme schreiben will, die tiefer in interne Abläufe eingreifen, kommt auf Dauer kaum ohne dieses Wissen aus. Die nächste Folge der Assemblerecke wird zeigen, wie man Betriebssystem-Routinen zeitsparend direkt im ROM aufruft, ohne daß es zu Problemen mit den unterschiedlichen CPC-Versionen kommt.

(Matthias Uphoff/cd)

Der schnelle BASIC-Wettbewerb

Gewinnen Sie den FAst BAsic COMpiler

Ein kleines Gewinnspiel soll Ihnen diesen Monat die Möglichkeit geben, Ihre selbst geschriebenen BASIC-Programme auf den CPCs um einiges im Tempo zu steigern.

Wir verlosen zehn FAst BAsic COMpiler, aber nur unter den Einsendungen, die unsere beiden Fragen zu allgemeiner Zufriedenheit beantworten.

Bitte geben Sie uns an, welchen CPC Sie besitzen, damit wir die Richtigkeit Ihrer Frage genauestens überprüfen können.

Die Fragen:

1. Wie lange benötigt Ihr CPC, um eine nichtstuende FOR-NEXT-Schleife 10000 (in Worten: zehntausend) Mal zu durchlaufen?
2. Was glauben Sie, welche Zeit für die gleiche Aktion benötigt wird, wenn das Programm mit unserem Compiler beschleunigt wurde?

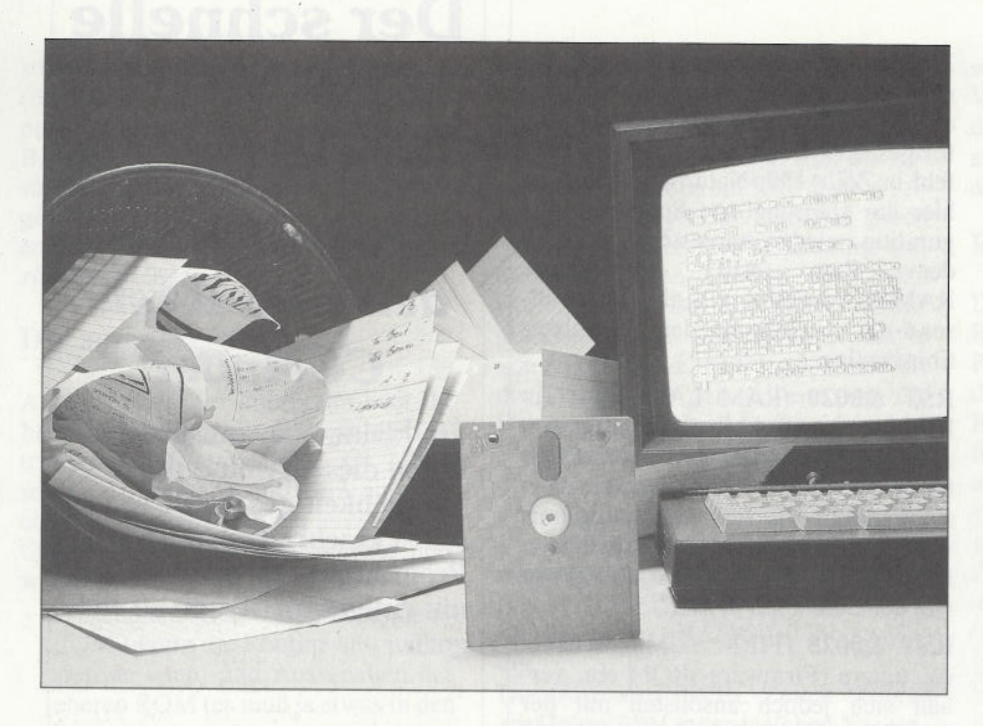
Schicken Sie Ihre Antworten an den

DMV-Verlag
Redaktion "PC Amstrad"
Stichwort "Compiler"
Postfach 250
3440 Eschwege

Einsendeschluß ist der 31. Januar 1990. Der Rechtsweg ist ausgeschlossen. Mitarbeiter des DMV-Verlages und deren Angehörige dürfen an der Verlosung nicht teilnehmen.

Und nun viel Geduld beim Messen und ebenso viel Glück beim Tippen.

(jf)



dBase verständlich

Einsteigerkurs für das bekannte Datenbanksystem

Nachdem wir uns in den letzten Teilen mit den Grundlagen von dBase beschäftigt haben, geht es heute mit dem Programmieren von kleinen Programmen weiter.

Als Vorbereitung müssen wir uns zunächst mit den "Suchvariablen" befassen. Es handelt sich dabei um normale, gespeicherte Variablen, wie sie auch in jeder anderen Programmiersprache vorkommen. Ihre Anzahl ist in dBase begrenzt: dBase II erlaubt insgesamt nur 64 Variablen (In III/III Plus sind es 256). Da sie oft zur Aufnahme von Werten dienen, die auch in den Dateifeldern vorkommen, können sie ebenfalls vom Typ Character numerisch oder logisch sein. Im Gegensatz zu den Feldern wird der Typ aber nicht durch Deklaration festgelegt (vgl. auch Pascal), sondern ergibt sich durch die Wertzuweisung (ähnlich BASIC).

Um eine Verwechslung mit Feldbezeichnungen zu vermeiden, wird empfohlen, daß die Bezeichnung von Speichervariablen mit einem M (Memory) beginnt. Wir werden den Namen dann in Kleinschreibung weiterführen. Die Wertzuweisung kann im Programm erfolgen. Wenn Mzahl den Wert 37.456 erhalten soll, so muß es in dBase II heißen

STORE 37.456 TO Mzahl

In dBase III/III Plus kann man auch (wie von anderen Sprachen zum Beispiel BASIC her gewohnt) schreiben: Mzahl = 37.456 Damit ist dann Mzahl als numerische Variable mit drei Dezimalstellen definiert. *Beachten Sie, daß durch STORE 37 TO Mzahl die Variable als Ganzzahl ohne Dezimalstellen definiert wird. Soll der Wert 37 betragen, aber mit zwei Dezimalstellen, so müssen Sie schreiben: STORE 37.00 TO Mzahl.* Weisen Sie einer Variablen eine Zeichenkette zu, so wird diese als Character definiert. Die Zeichenkette muß (wie auch in BASIC) in Anführungszeichen eingeschlossen werden.

STORE "Dies ist eine Zeichenkette" TO Mzeichen

Mit Speichervariablen lassen sich Operationen ausführen, wie Sie es von BASIC her kennen. Für numerische Variablen gelten die arithmetischen Operatoren

"+", "-", "*", "/"

("**" bzw. "↑" für Potenzieren gibt es nur in III/III Plus). Bei Zeichenketten ist mit "+" eine Zusammenfügung (Concatenation) möglich, das heißt, es können mehrere Teile zu einem String verbunden werden. Den wichtigen Substring-Operator "$" haben wir bereits kennengelernt. Er liefert als Ergebnis den logischen Wert WAHR oder FALSCH.

"and"$"Kandelaber" ist WAHR.
"und"$"Kandelaber" ist FALSCH ebenso wie "And"$"Kandelaber".

Um Speichervariablen auszugeben, kann das Kommando "?" benutzt werden.

"? Mzahl" gibt den Wert von Mzahl und "? Mzeichen" die gespeicherte Zeichenkette aus.

Die parallele Druckerausgabe erfolgt mit SET PRINT ON. Das Fragezeichen allein bewirkt nur einen Zeilenvorschub. Es ist sehr wichtig für uns, daß die Werte für die Speichervariable eingegeben werden können. Dazu dient zunächst das Kommando INPUT. Das funktioniert ähnlich wie bei BASIC.

INPUT TO Mvariable

(das TO nicht vergessen!)

bewirkt, daß ein Doppelpunkt vor dem Cursor erscheint. Dann können Sie den Wert eingeben. Eine Zeichenkette muß wieder in Anführungszeichen stehen. Wie bei BASIC können Sie vor der Eingabe einen Erläuterungstext ausgeben:

INPUT "Betrag der Prämie" TO Mpraemie

INPUT "Vorname des Kunden" TO Mvorname

Beachten Sie bitte, daß Sie innerhalb einer Zeichenkette Sonderzeichen wie die Umlaute "Ä" oder "Ü" verwenden können. Dies ist bei einer Variablenbezeichnung jedoch nicht möglich! Zwei Nachteile hat dieses Eingabekommando: Bei Zeichenketten (Dateiverarbeitung) müssen Sie immer die lästigen Anführungszeichen verwenden, und Sie können eine Eingabe nicht "überspringen". Das heißt, wenn Sie zum Beispiel den Vornamen offenlassen wollen und nur RETURN eingeben, erhalten Sie eine Fehlermeldung und das Programm fordert die Wiederholung. Beide Nachteile vermeidet das Kommando ACCEPT, das dann aber nur für eine Charactervariable verwendet werden kann.

ACCEPT "Name des Kunden" TO Mname

bewirkt wieder die Ausgabe des Erläuterungstextes mit anschließendem Doppelpunkt vor dem Cursor. Jetzt geben Sie den Namen ohne Anführungszeichen ein und nach dem RETURN ist dieser als Zeichenkette gespeichert. Nur RETURN erzeugt eine leere Zeichenkette. Wir werden daher ausführen. Das Problem ist von grundsätz-

schließlich ACCEPT verwenden. Zunächst aber sollten Sie sich mit der Handhabung von Speichervariablen eingehend vertraut machen. Wir kommen jetzt zu der eigentlichen Programmierung von Suchoperationen, wobei wir regen Gebrauch von Speichervariablen machen.

Die Programmierung von Abfragen

Betrachten wir nun das Programm zur Kundensuche in Listing 1: Programmzeilen, die mit einem Stern beginnen, sind Erläuterungszeilen, die nicht ausgeführt werden (REM in BASIC). SET TALK OFF schaltet die Systemmeldungen ab. Bei einer zügigen Programmausführung würde dieses (sonst sehr nützliche) "Geschwätz" nur stören. ERASE löscht den Bildschirm, RELEASE ALL löscht alle bereits vorher mit Werten definierten Variablen im Speicher. (Das sollte man vorsichtshalber immer tun.) Jetzt folgen die drei Eingaben für den gesuchten Kunden: Name, Vorname, Wohnort; sie werden drei Speichervariablen zugewiesen. Ein Zeilenvorschub ("?") sorgt für den nötigen Abstand vor der Ausgabe der gewünschten Datensätze, die – nach dem Öffnen der Kundendatei – mit dem Kommando DISP FOR erfolgt. CLEAR schließt die Datei, das "Geschwätz" wird wieder eingeschaltet und RETURN muß als Kommando eingegeben werden, da dies eine Unterroutine beendet. Der Tastendruck ist hier nicht gemeint.

Geben Sie jetzt dieses Programm ein. Sie können dies mit einem Textverarbeitungssystem machen, dabei sollte das Programm in reinem ASCII-Code abgespeichert werden. Wenn WordStar verwendet wird, sollte es mit der N-Option abgespeichert werden. Unter CP/M speichern Sie es als Datei KSUCHEN1.CMD, unter MS-DOS als KSUCHEN1.PRG. dBase enthält einen eingebauten Programmeditor, der zwar nicht so komfortabel ist wie ein echter Texteditor, aber für einfache Beispiele ausreicht. Er wird aufgerufen mit MODIFY COMMAND. Sie können gleich den Programmnamen mit eingeben, ansonsten werden Sie dazu aufgefordert. Geben Sie nur KSUCHEN1 ein, die Namenserweiterung (CMD oder PRG) wird von dBase besorgt. Tabelle 1 gibt Ihnen die Tastenkombinationen für die Cursorsteuerung. Nach Eingabe von CTRL-W wird das Programm gespeichert.

Um ein Programm unter dBase zu starten, benutzen Sie das Kommando DO, also DO KSUCHEN1. Wenn Sie bei der Eingabe keinen Fehler gemacht haben, muß alles prinzipiell so ablaufen, wie es früher bei der "Handsuche" der Fall war. Die Handhabung ist aber wesentlich bequemer, denn unser Programm enthält ja schon alle erforderlichen Kommandos, die wir früher von Hand immer wieder einzeln eingeben mußten. Testen Sie das Programm mit allen möglichen Eingaben. Versuchen Sie auch "Leereingaben."

Zur Ausgabe der gesuchten Sätze benutzt es DISP mit FOR-Bedingung. (Man kann natürlich auch LIST verwenden.) Auch einen solchen Ablauf können wir in dBase programmieren (Listing 2). Wir sehen drei neue Kommandos.

LOCATE FOR <Bedingung>
sucht vom TOP (Beginn) der Datei aus den ersten Satz mit der Bedingung. Wird er gefunden, so steht der interne Satzzeiger entsprechend. Wenn nicht, so steht der interne Satzzeiger auf BOTTOM (Ende), und EOF (End Of File) ist gesetzt.

DO WHILE
ist das "Falls"-Kommando,

CONTINUE
setzt die Suche mit der bei LOCATE angegebenen Bedingung fort, und zwar mit dem auf den jeweiligen Satzzeiger folgenden Satz. Die DO-WHILE-Bedingung muß "eingeklammert" werden (vgl. Pascal). ENDDO ist die "schließende Klammer". Testen Sie dieses Programm, und Sie stellen fest, daß es genauso arbeitet wie Listing 1.

Mit dieser programmierten Schleife können wir an Stelle einer einfachen Satzausgabe mehrere verschiedene Operationen ausführen. Denken Sie zunächst an unser erstes Suchbeispiel. Der Ablauf ist in Abbildung 1 dargestellt. Abbildung 2 schließlich zeigt den Ablauf für unser zweites Suchbeispiel. Wenn Sie sich diese Abläufe genau ansehen, werden Sie feststellen, daß sie etwas anders sind, als die früher von Ihnen "per Hand" ausgeführten. Dort haben Sie zunächst **alle** Informationen aus einer Datei entnommen (Kundennummern) – mußten sich diese merken – und dann konnten Sie die gesamten dazu gehörenden Informationen aus der zweiten Datei entnehmen. Hier hingegen wird abwechselnd zwischen beiden Dateien hin- und hergeschaltet. Dieses Verfahren ist wesentlich eleganter.

Im ersten Beispiel erhalten Sie die gesuchten Kunden aufgelistet, und unter jedem Kunden stehen entweder seine sämtlichen Verträge oder die Verträge der gesuchten Versicherungsart. Im zweiten Beispiel sind alle in Frage kommenden Kunden der Reihe nach aufgelistet. Bei der Programmierung dieser Abläufe ist noch eine Schwierigkeit zu überwinden. Wie schaltet man von einer Datei zur anderen? Mit dem Kommando USE geht das nicht. Wenn Sie zum Beispiel mit

USE VERSGES
diese Datei öffnen und später
USE KUNDEN
eingeben, so wird zwar KUNDEN geöffnet, aber auch VERSGES geschlossen. Leider ist damit dann der Satzzeiger in VERSGES "futsch". Beim Wiederöffnen wird er neu an den Anfang (TOP) gesetzt, und Sie können eine begonnene Suche dann nicht mehr fortführen. Das Problem ist von grundsätzlicher Art. Ein Datenbanksystem muß

```
-----------------------------------------------------------
Tastenkombination     Cursorfunktion

===========================================================

CTRL - E  nach oben
CTRL - X  nach unten
CTRL - S  ein Zeichen zurück
CTRL - D  zum nächsten Zeichen
CTRL - A  ein Wort zurück
CTRL - F  zum nächsten Wort
CTRL - Y  Feld löschen
CTRL - V  Einfügemodus EIN/AUS ("Wechselschalter")
CTRL - G  Zeichen unter Cursor löschen
DELETE    Zeichen links vom Cursor löschen
CTRL - T Cursorzeile löschen
         restliche Zeilen rücken auf
CTRL - N Neue Zeile einfügen
CTRL - C Bildschirm nach unten "scrollen"
CTRL - R Bildschirm nach oben "scrollen"
CTRL - B Cursor an das Zeilenende
CTRL - Z Cursor an den Zeilenanfang
CTRL - Q Operation MODI COMM wird beendet
die geänderte Datei wird nicht gespeichert
CTRL - W Operation MODI COMM wird beendet
     die geänderte Datei wird gespeichert

-----------------------------------------------------------
```

Tabelle 1: Cursorsteuerung für Modify Command

```
Wähle Datei KUNDEN
Suche Satz mit angegebener Bedingung.
Falls nicht Ende der Datei
    (Gib den Satz aus,
    Wähle Datei VERSGES,
    gib Sätze aus
        für gleiche Kundennummer
            (AND gegebene Vers. Art),
    Wähle Datei KUNDEN,
    fahre mit Suche fort),
Ende "Falls".
```

Abbildung 1: Suchschleife für Suchbeispiel 1

es gestatten, mehrere Dateien gleichzeitig "offenzuhalten", so daß man von einer zur anderen beliebig umschalten kann, wobei der jeweilige Satzzeiger in den Dateien seinen Platz behält. Jede Datei besitzt dann einen eigenen "Arbeitsbereich", unabhängig von den anderen. dBase II erlaubt nur zwei Arbeitsbereiche (für zwei Dateien). In III/III Plus können es bis zu zehn sein. *Das ist eine Frage des zur Verfügung stehenden RAM-Speichers. Dem für CP/M konzipierten dBase II fehlte einfach der Speicherplatz.* Unter dBase II heißen die Arbeitsbereiche PRIMARY und SECONDARY, unter III/III Plus einfach 1 bis 10 (oder auch A bis J). Man muß zunächst die Dateien den einzelnen Arbeitsbereichen "zuteilen". Das geschieht mit den Kommandos SELECT und USE.
SELECT PRIMARY
USE KUNDEN
SELECT SECONDARY
USE VERSGES
teilt KUNDEN dem Bereich 1 und VERSGES dem Bereich 2 zu. *In III/III Plus heißt es dann einfach SELECT 1, SELECT 2, usw. bis SELECT 10 oder auch SELECT A, SELECT B usw.* Das "Umschalten" erfolgt ebenfalls mit dem SELECT-Kommando. Sie können das Verfahren noch etwas abkürzen, denn die **erste** geöffnete Datei wird automatisch immer dem "Hauptarbeitsbereich" 1 (PRIMARY) zugewiesen. Jetzt werden Sie die in Listing 3 und 4 programmierten Abläufe für unsere Suchbeispiele verstehen. Im Listing 3 wird auch die Versicherungsart abgefragt. Wenn Sie nur RETURN eingeben, erhalten Sie alle Verträge. Ebenso können Sie bei der Kundensuche Ort und auch Vornamen "ausblenden". Probieren Sie alles aus.
Noch etwas zur Syntax der Suche nach den Sätzen mit gleichem Feld (hier KUNDNR) in den verschiedenen Arbeitsbereichen. Bei dBase II ist P.KUNDNR das Feld der Datei im PRIMARY- und S.KUNDNR das Feld im SECONDARY-Bereich. Bei III/III Plus kann der Arbeitsbereich durch seinen Buchstaben bezeichnet werden (keine Zahl). Es kann aber auch der Dateiname angegeben werden. Statt des Punktes steht aber immer ein Pfeil (->). Also A->KUNDNR = B-> KUNDNR oder auch KUNDEN-> KUNDNR = VERSGES->KUNDNR
Es gibt aber noch wesentlich mehr davon. Bei den bisher besprochenen Suchkommandos wird jeder einzelne Satz einer Datei (vom ersten bis zum letzten) nach dem Suchbegriff durchforscht. Es spielt dabei keine Rolle, in welcher Anordnung sich die Sätze befinden. Läßt sich durch eine bestimmte Ordnung der Suchvorgang beschleunigen? Mit dieser Frage wollen wir uns jetzt beschäftigen.

Ordnung in den Dateien

Als eine brauchbare Ordnung bietet sich an, die Datei nach dem Inhalt des Feldes mit dem Suchbegriff zu sortieren. Man nennt das dann "Schlüsselfeld" und den gesuchten Inhalt das "Schlüsselwort". Bei der Datei KUNDEN wäre es zum Beispiel die Kundennummer oder der Kundenname. Jedes Feld einer Datei kann Schlüsselfeld sein. Man unterscheidet aber "Primärschlüssel" und "Sekundärschlüssel". Der erste ist immer eindeutig: Zu jedem Wert gehört (höchstens) ein Datensatz. (Beispiel: Kundennummer.) Zu einem Wert des Sekundärschlüssels können jedoch mehrere Sätze gehören (Beispiel: Kundenname).
Das Sortieren ist ein aufwendiger Prozeß, denn die Sätze müssen entsprechend der Sortierfolge umgespeichert werden. Die physikalische Reihenfolge der Sätze wird also geändert. Das Ergebnis steht in einer neuen Datei. dBase hat das Kommando SORT, mit dem man eine Datei nach einem gewünschten Feld sortieren kann. Das Verfahren hat aber erhebliche Nachteile. Der Prozeß ist recht langsam (vor allem bei dBase II). Außerdem ist die Ordnung wieder futsch, wenn die Datei durch neue Sätze erweitert wird. Dann muß neu sortiert werden. Bringt uns die Ordnung nun eine wesentlich beschleunigte Suche? In einer ungeordneten Datei müssen ja jedesmal **alle** Sätze durchsucht werden.
In einer geordneten Datei können Sie auch nur die **halbe** Zeit einsparen. Wenn Sie nämlich in einer nach Namen sortierten Datei nach "Abelmann" suchen, so können Sie bald aufhören, bei "Zitzewitz" müssen Sie fast die ganze Datei absuchen, im Mittel also die halbe Datei. Nun, das war es schon wieder, und wir hoffen, das Sie auch beim nächsten Mal wieder dabei sind. Denn der nächste Teil wird unser Abschluß dieses Kurses sein, wobei es auch noch eine Menge an Literaturhinweisen gibt.

(Hans-Georg Sanner/cd)

```
Wähle Datei VERSGES
Suche Satz mit angegebener Bedingung
    (Gesellschaft AND Vers.Art)
Falls nicht Ende der Datei
    (Wähle Datei KUNDEN,
    Gib Sätze aus
        für gleiche Kundennummer,
    Wähle Datei VERSGES,
    fahre mit Suche fort),
Ende "Falls".
```

Abbildung 2: Suchschleife für Suchbeispiel 2

```
****************** KSUCHEN1.CMD *********************************
************** Kundensuche VERSION 1  ***************************
*
*
SET TALK OFF
ERASE
RELEASE ALL
*
ACCEPT "Name des Kunden  " TO Mkuname
ACCEPT "Vorname des Kunden  " TO Mvorname
ACCEPT "Ort (Postleitzahl)  " TO Mort
*
?
*
USE KUNDEN
DISP OFF FOR Mkuname$NAME .AND. Mvorname$VORNAME .AND. Mort$ORT
*
*
CLEAR
SET TALK ON
RETURN
```

```
****************** KSUCHEN2.CMD *********************************
************** Kundensuche VERSION 2  ***************************
*
*
SET TALK OFF
ERASE
RELEASE ALL
*
ACCEPT "Name des Kunden  " TO Mkuname
ACCEPT "Vorname des Kunden  " TO Mvorname
ACCEPT "Ort (Postleitzahl)  " TO Mort
*
?
*
USE KUNDEN
*
LOCATE FOR Mkuname$NAME .AND. Mvorname$VORNAME .AND. Mort$ORT
DO WHILE .NOT. EOF
   DISP OFF
   CONTINUE
ENDDO
*
CLEAR
SET TALK ON
RETURN
```

```
****************** SUCHBSP1.CMD *********************************
**************** Suchbeispiel 1 ********************************
*
*
SET TALK OFF
ERASE
RELEASE ALL
*
ACCEPT "Name des Kunden  " TO Mkuname
ACCEPT "Vorname des Kunden  " TO Mvorname
ACCEPT "Ort (Postleitzahl)  " TO Mort
*
?
ACCEPT "Versicherungsart  " TO Mversart
*
SELECT PRIMARY
USE KUNDEN
SELECT SECONDARY
USE VERSGES
*
SELECT PRIMARY
LOCATE FOR Mkuname$NAME .AND. Mvorname$VORNAME .AND. Mort$ORT
DO WHILE .NOT. EOF
   DISP OFF
   ?
   SELECT SECONDARY
         DISP OFF FOR S.KUNDNR=P.KUNDNR .AND. Mversart$VERSART
   ?
   SELECT PRIMARY
   CONTINUE
ENDDO
CLEAR
SET TALK ON
RETURN
```

```
***************** SUCHBSP2.CMD *********************************
**************** Suchbeispiel 2 ********************************
*
*
SET TALK OFF
ERASE
*
SELECT PRIMARY
USE KUNDEN
SELECT SECONDARY
USE VERSGES
*
? "Ausgabe der Kunden f. best. Vers.-Art bei best. Gesellschaft"
?
ACCEPT "Vers.-Art  " TO Mversart
ACCEPT "Gesellschaft  " TO Mges
?
*
LOCATE FOR Mges$GES .AND. Mversart $VERSART
DO WHILE .NOT. EOF
   SELECT PRIMARY
   DISP OFF for P.KUNDNR=S.KUNDNR
   SELECT SECONDARY
   CONTINUE
ENDDO
CLEAR
SET TALK ON
RETURN
```

Listing Suchbeispiele



100, – DM für 1 kByte

Die Herausforderung

“Wo nicht die Zeit ist, fasse dich kurz“, bemerkte schon Sophokles, der zugegeben in einer wesentlich gemäßigteren Epoche lebte, aber wohl schon damals ahnte, mit welchen akuten Zeitdefiziten wir uns in unserer schnellebigen Gegenwart herumzuschlagen haben. Wer hat denn heute noch etwas Zeit übrig? Niemand, und genau deshalb fassen sich alle immer kürzer. Auch in der Programmierung, wie Sie an folgenden Beispielen erkennen werden.

Slide

Nur 84 Byte waren notwendig, um dieses Maschinenprogramm zu erstellen! Es sorgt dafür, daß Grafiken auf dem CPC nach Art eines Dia-Projektors in den aktuellen Bildschirm eingeblendet werden. Das aktuelle Bild wird über den rechten Bildschirmrand aus dem Blickfeld geschoben, während das neue von links hereingezogen wird.

Diese Maschinenroutine ist frei im Speicher verschiebbar, so daß es keine Probleme gibt, das Programm in eigene Werke einzubauen. In unserem Fall liegt es an der Adresse &A000.

Laden Sie Slide und starten Sie es. Aktiviert wird die Routine mit dem Befehl CALL <Adresse>, <Anfang>.

“Adresse“ bezeichnet die Position im Speicher, wo das Programm abgelegt worden ist. “Anfang“ ist die Startadresse des Bildes, das hereingezogen werden soll.

Sollte dem CALL-Befehl kein Parameter mitgegeben werden, so geschieht nichts, da in diesem Fall ein sofortiger Rücksprung zum BASIC erfolgt.

Der Videochip 6845 ermöglicht ein Beeinflussen der Synchronisationssignale, die zusammen mit dem Videosignal an den Monitor geschickt werden. Das Register 2 des 6845 ist hierbei für die horizontale Richtung zuständig und hat im Normalfall den Wert 46. Der gesamte Bildschirm samt Rahmen und (nicht sichtbarer) Austastlücke hat eine Breite von 64 Spalten, so daß Werte von 0 bis 63 möglich sind.

Die Routine nun vermindert den Ausgangswert 46, bis dieser auf 34 geschrumpft ist. Dann liegt der rechte Rand des Bildschirms genau in der Mitte des nicht sichtbaren Bereichs. Übrigens sollte vorher nicht gescrollt werden, da sich sonst die Bildschirmadressen verschieben und Teile wieder im sichtbaren Bereich liegen. Nun wird spaltenweise das neue Bild einkopiert, der Bildschirm wiederum verschoben und so fort, bis das gesamte Bild kopiert ist und nun am linken Rand erscheint.

Eine Verzögerungsroutine sorgt dafür, daß das träge menschliche Auge überhaupt etwas von der ganzen Aktion mitbekommt. Erreicht Register 2 den Wert 0, so wird es beim nächsten Mal auf 63 gesetzt; wird schließlich der Ausgangswert 46 erreicht, so übergibt das Programm an BASIC zurück.



(Martin Sachenbacher/jf)

Sterne

Zu den üblichen zwei Möglichkeiten, Sterne zu sehen (abends am Himmel oder nach einer Rempelei mit einem üblen Rüpel) gesellt sich nun eine weitere: am CPC nach dem Start des vorliegenden Programms.

Sterne malt Ihnen das Gelbe vom Nachthimmel auf einfachste Weise. Sie laden das Programm und werden aufgefordert, drei Werte einzugeben.

Die ersten beiden Werte stehen für Außen- bzw. Innenradius, der dritte bestimmt die Anzahl der Zacken. Das Prinzip, nach dem nun Ihr Stern gezeichnet wird, ist relativ einfach: Die Werte der Radien bestimmen zwei Kreise, auf denen regelmäßig Punkte gesetzt werden. Die Anzahl der Punkte ist dabei natürlich gleich der Anzahl der Zacken. Sind die Punkte gesetzt, werden alle des Außenradius' mit jedem des Innenradius' verbunden.

Danach verringert das Programm beide Radien und zeichnet erneut alle Verbindungslinien der so auf kleineren Kreisen postierten Punkte. Dieser Vorgang wird sechs mal durchlaufen, wobei stets eine neue Farbe gewählt wird. Ein farbenfroher Stern entsteht, der bei etwas Experimentierfreude in interessante Himmelsbrüder verwandelt werden kann.

Um Ihre Kreationen sichern zu können, reicht es aus, dem Listing zwei Zeilen zuzufügen:

```
35 INPUT "Bildname : ",n$
190 if n$<>"" then save n$,b,&c000,&4000
```

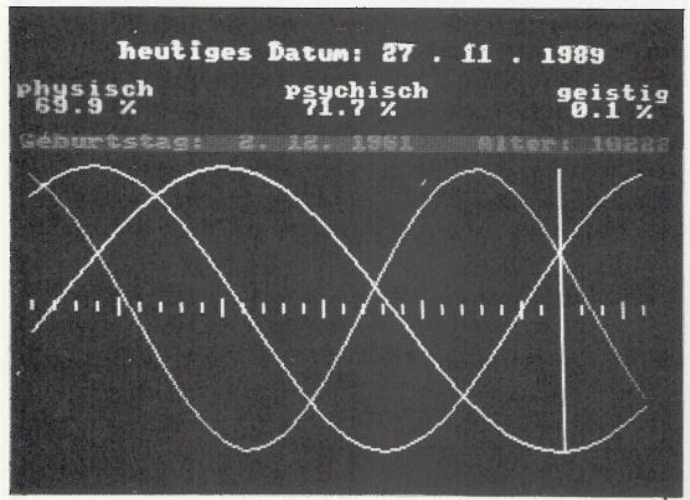
Sie sind dadurch in der Lage, einen Namen zu bestimmen, mit dem das entstandene Bild auf Diskette abgespeichert wird. Geben Sie keinen Namen ein, so wird auch nichts gesichert.

Durch die Einfachheit der Programmierung laden diese Zeilen förmlich zu Erweiterungen ein. Durch Veränderung der Farbwerte ist es beispielsweise denkbar, Ansätze eines sternförmigen Lauflichts auf den Monitor zu bekommen. Probieren Sie folgende Variante:

```
5 ink 2,15,6 : ink 3,6,15
55 if fa=2 then fa=3 else fa=2
90 plot x,y,fa
130 move x,y: draw x1,y1
```

Starten Sie nach den Änderungen das Programm mit den Parametern Außenradius=200, Innenradius=50 und Teilung =10, Sie werden überrascht sein.

(Jürgen Krämer/jf)



Der Biorhythmus erklärt Ihre Depressionen und Hochstimmungen auf mathematische Weise

Speerwurf (der Olympiade erster Teil)

Wer kennt schon Summer Games? Wozu auch, gibt es doch den ultrakurzen Speerwurf für alle CPC-Rechner. Sie halten das für unglaublich? Nach dem Abtippen dieses ersten Teils der 1 kByte-Olympiade werden Sie anderer Ansicht sein. Nehmen Sie sich Ihren Joystick zur Hand und rütteln Sie Ihrem Speerwerfer die nötige Kraft in den Wurfarm.
Wichtig ist die zeitige Trennung vom Wurfgeschoß (der Hindernislauf soll Gerüchten zufolge in Arbeit sein), will man doch erleben, an welcher Marke der Speer zurück zur Erde findet. Das jeweils beste Ergebnis wird am oberen Bildschirmrand angezeigt, – ein ungeheurer Ansporn für alle Speerwerfer.

(Christoph Möller/jf)

Hürdenlauf (der zweite Streich)

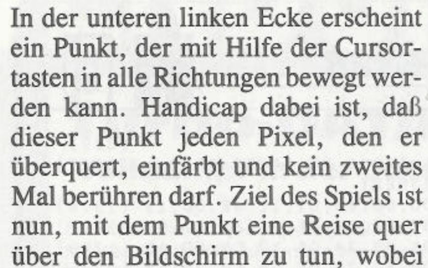
Und weiter geht's mit dem Fitneßtraining vor dem Bildschirm. Nach der ersten technischen Disziplin (Speerwurf) folgt in unserer Miniolympiade die zweite: der Hürdenlauf. Dieser erscheint auf dem Bildschirm wesentlich ungefährlicher zu sein als in der groben Wirklichkeit, wo während des Laufens permanent in den Weg gestellte Barrikaden den Laufwilligen dazu veranlassen, riskante Sprünge zu tätigen. Beim kleinsten Hängenbleiben landet der Sportler so auf der Nase, was mitunter schmerzlich sein kann.
Unser 1 kByte Programm verfolgt dagegen streng den Trend "Sport ist gesund" und behütet so die Hürdenläufer vor schmerzhaften Hautabschürfungen.
Nach dem Ladevorgang ertönen drei Pieptöne, worauf das Rennen beginnt. Heftiges Auf- und Abbewegen des Joysticks lassen die Spielfigur auf Touren kommen. Die Hürden werden durch rechtzeitiges Drücken auf den Feuerknopf gemeistert. Am Ende des Parcours wird die Zeit genommen, also nichts wie los!

(Steffen Rau/Jens Köplinger/jf)

Misthaufen

Bei Misthaufen handelt es sich um ein Spiel, das sowohl Geist als auch Geschicklichkeit vom Spieler abverlangt. Das Programm meldet sich mit einer großen Anzahl von Zahlen.

Ein besonders gelungenes Exemplar eines Sterns lädt zum Experimentieren mit unserem Sternprogramm ein



In der unteren linken Ecke erscheint ein Punkt, der mit Hilfe der Cursortasten in alle Richtungen bewegt werden kann. Handicap dabei ist, daß dieser Punkt jeden Pixel, den er überquert, einfärbt und kein zweites Mal berühren darf. Ziel des Spiels ist nun, mit dem Punkt eine Reise quer über den Bildschirm zu tun, wobei als Etappen die zu Beginn erwähnten Ziffern dienen. Erste Station ist die Eins, zweite die Zwei und so weiter.
Der Bildschirm füllt sich schnell mit eingefärbten Punkten, so daß die Wege mit der Zeit arg eingeschränkt werden. Da ist Übersicht gefragt. Planen Sie Ihre Routen sorgfältig, damit Sie nicht in einer selbstgebauten Sackgasse enden.

(Sascha Best/jf)

Biorhythmus

Die Theorie des Biorhythmusses geht darauf zurück, daß mit der Geburt die Zustände des Geistes, der Seele und des Körpers in bestimmten Zyklen mal mehr positiv, mal mehr negativ anzusehen sind. Diese Zyklen verlaufen nicht parallel, lassen sich aber mathematisch relativ einfach berechnen. Vorliegendes Programm liefert eine grafische Auswertung dieser Zyklen, dazu eine prozentuale Darstellung des aktuellen Tages und das Alter der Testperson in Tagen.
Der Benutzer muß lediglich das Datum seines Geburtstages und das des aktuelle Tages eingeben. Der Rest wird vom Computer erledigt.
Er zeichnet die drei Kurven für die Zustände des Geistes, der Seele und des Körpers auf den Bildschirm und lädt den Benutzer zur Interpretation ein.
Keine Angst, es ist fast sicher, daß stets eine der Kurven im Minusbereich umherirrt. Das sollte kein Grund zur Beunruhigung sein. Glaubt man nämlich der Theorie des Biorhythmusses, so befindet man sich nur dann in einem echten Tief, wenn alle Kurven den unteren Bildschirmrand berühren, aber auch in einem solchen Stadium sollten Sie sich nicht allzu großen Depressionen hingeben. Außerdem sehen Sie ja schon einige Tage vorher, inwieweit sich Ihr Rhythmus entwickelt. An kritischen Tagen lassen Sie das Programm nach dem Motto "Was ich nicht weiß, macht mich nicht heiß" einfach in der Schublade.

(Sönke Ostertun/jf)

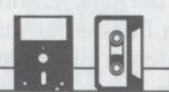
Die Bedingungen

Hier die Bedingungen unseres Wettbewerbs: Wir suchen komplette Programme (bitte ohne Steuerzeichen!), die funktionsfähig möglichst auf allen drei CPCs laufen (was nicht heißen soll, daß wir bei besonders guten keine Ausnahme machen), und die nicht länger als 1 kByte sind (Anzeige auf dem Datenträger gilt!). Der Preis für solche Programme beträgt 100 DM. Senden Sie uns Ihre Werke (maximal zwei Programme!) auf Datenträger (Diskette/Kassette) mit Programmbeschreibung (möglichst in ASCII-Format mit auf dem Datenträger) zu und zwar an den

DMV-Verlag
Postfach 250
Stichwort 100,– DM
3440 Eschwege



für 464-664-6128



```
1 REM Slide                                    [329]
2 MEMORY &9FFF:FOR i=&A000 TO &A053:READ a [3317]
$:POKE i,VAL("&"+a$):NEXT
3 DATA B7,C8,DD,6E,00,DD,66,01,11,4E,00,19 [1576]
4 DATA 16,C0,01,02,BC,ED,49,0E,2D,3E,22,91 [1901]
5 DATA 30,05,3E,3A,91,30,24,06,19,C5,ED,A0 [1793]
6 DATA ED,A8,7C,C6,08,67,7A,C6,08,57,30,F2 [2345]
7 DATA 01,50,C0,09,EB,09,EB,C1,10,E7,79,01 [1919]
8 DATA 2E,F8,09,EB,09,EB,4F,CD,19,BD,06,BD [2237]
9 DATA 79,ED,79,FE,2E,C8,3D,E6,3F,4F,18,C1 [2313]
```

Listing Slide

```
5 RAD                                          [301]
6 MODE 1                                       [506]
10 INPUT "Aussenradius: ",e                   [1801]
20 INPUT "Innenradius : ",f                   [1538]
30 INPUT "Teilung     : ",n                   [2274]
40 CLS:r=e:r1=f                                [365]
50 FOR c=7 TO 2 STEP -1                       [1212]
60 FOR t=1 TO n                                [821]
70 x=320+r*COS((2*PI/n)*t)                    [2141]
80 y=200+r*SIN((2*PI/n)*t)*0.9                [2556]
90 PLOT x,y,c                                  [771]
100 FOR v=1 TO n                              [1268]
110 x1=320+r1*COS((2*PI/n)*v)                 [1617]
120 y1=200+r1*SIN((2*PI/n)*v)*0.9             [1682]
130 MOVE x,y:DRAW x1,y1,c                     [1534]
140 NEXT                                       [358]
150 NEXT t                                     [360]
160 r=r*0.7                                    [530]
170 r1=r1*0.7                                 [1042]
180 NEXT c                                     [381]
```

Listing Sterne

```
10 DEG:BORDER 3:INK 0,3:INK 1,26:INK 2,0:S [16282]
YMBOL 255,3,60,48,112,56,56,40,40:MODE 1:P
RINT"Kraft:","Rekord:"v"Weite:":l=1:TAG:FO
R k=116 TO 640 STEP 25:MOVE k,198:DRAWR 0,
-4,2-((k=116)+1):MOVER -32,-4-(k MOD 2)*16
:PRINT i;:i=i+5
20 NEXT:WHILE INKEY(74)<>0:WEND:FOR k=0 TO [4569]
 545:IF NOT INKEY(74) AND l THEN l=0:r=1:g
=g+1
30 IF NOT INKEY(75) AND r THEN r=0:l=1:g=g [1905]
+1
40 IF NOT INKEY(76) THEN g=g-25+k/20:TAGOF [4601]
F:SOUND 1,0,4,15,,,1:GOTO 70
50 MOVE 0,370:CALL &BD19:DRAWR g,0,3:DRAWR [4011]
 -g,0,0:g=g-0.2:IF g<0 THEN g=0
60 PLOT -10,-10,1:MOVE k/5,216:PRINT CHR$( [8186]
255);:k=k+g/10:NEXT:TAGOFF:LOCATE 15,20:PR
INT"UNGUELTIG !":g=0:i=0:CALL &BB18:GOTO 1
0
70 FOR h=0 TO 5*g STEP 4                      [1324]
80 a=(g+h/32)*SIN(h/(5*g/180)):s=k/5-5:MOV [14033]
E h+s,200+a:z=204+((g+(h+20)/32))*SIN((h+1
6)/(5*g/180)):DRAW h+s+20,z,1:MOVE h+s,200
:DRAWR 20,0,2:MOVER -22,0:DRAWR 4,0,0:MOVE
 h+s,200+a:DRAW h+s+20,z,0:IF z<=200 AND k
>20 THEN q=h:h=5*g
90 NEXT:MOVE q+s+20,z:DRAWR -20,8,1:g=ROUN [5207]
D((q+s-96)/5,2):LOCATE 33,1:PRINT g:IF g>v
 THEN v=g
100 WHILE INKEY(76)<>0:WEND:g=0:i=0:GOTO 1 [2919]
0
```

Listing Speerwurf

```
10 MODE 1:INK 3,3:PAPER 3:CLS:BORDER 3:PAP [3184]
ER 0:WINDOW 15,25,10,15:CLS
20 LOCATE 1,6:PRINT STRING$(11,143);:PEN 2 [5114]
:LOCATE 3,5:PRINT CHR$(248)
30 FOR g=1 TO 600:PRINT CHR$(-7*((g MOD 15 [2810]
0)=0));:NEXT:t=TIME
40 j=1:x=384:TAG:y=190                         [859]
50 jo=0:PLOT 256,y,2:PRINT CHR$(249+j);:MO [9797]
VE 258,y+2:DRAWR 12,0,0:MOVER 0,2:DRAWR-12
,0:WHILE(jo AND 3)<>j AND(jo AND 16)<>16:j
o=JOY(0):WEND:j=3-j
60 IF(jo AND 16)=16 THEN IF up=0 THEN up=4 [3238]
:GOTO 50 ELSE IF jo-16=0 THEN 50
70 x=x-4:PLOT x,190,2:PRINT CHR$(148);      [2270]
80 IF x=260 AND y<198 THEN FOR w=1 TO 200: [3686]
NEXT:GOTO 50
90 IF x=224 THEN MOVE x,190:PRINT" ";:x=38 [2659]
4:r=r+1:IF r>5 THEN 120
100 y=y+up:IF y>204 THEN up=-4 ELSE IF y<1 [3341]
90 THEN up=0:y=190
110 GOTO 50                                    [384]
120 tg=TIME-t                                  [144]
130 tg=tg/300:sec=ROUND(tg,2)                 [1332]
140 WINDOW SWAP 0,1:TAGOFF:PEN 1:LOCATE 9, [5554]
20:PRINT"Sie brauchten"sec"Sekunden"
150 FOR t=1 TO 5000:NEXT:WHILE INKEY$<>"": [2126]
WEND
```

Listing Hürdenlauf

```
10 MODE 1:DIM s(40),z(40)                     [1171]
20 PEN 3:GRAPHICS PEN 3                       [1937]
30 x=2:y=2                                     [803]
40 DATA 18,5,8,13,26,10,7,6,16,16,31,3,16, [10763]
10,28,16,35,10,10,2,11,19,22,6,31,19,10,9,
37,2,3,20,2,9,21,14,31,6,19,20,2,3,19,2,3,
15,35,16,30,12,25,20,12,13,26,2,12,5,37,20
,37,6,7,17,2,12,20,10,7,4,22,17,36,13,15,7
,9,22,30,22
50 FOR a=1 TO 40:READ s(a),z(a):LOCATE s(a [3081]
),z(a):PRINT a:NEXT
60 PLOT 0,0:DRAW 639,0:DRAW 639,399:DRAW 0 [2301]
,399:DRAW 0,0
70 FOR a=1 TO 40                               [880]
80 PEN 2:LOCATE s(a)+1,z(a):PRINT MID$(STR [3351]
$(a),2,LEN(STR$(a)))
90 IF INKEY(0)=0 THEN r=1:y=y+2              [1634]
100 IF INKEY(8)=0 THEN r=1:x=x-2             [1758]
110 IF INKEY(1)=0 THEN r=1:x=x+2             [1195]
120 IF INKEY(2)=0 THEN r=1:y=y-2              [844]
130 IF r>0 THEN IF TEST(x,y)=3 THEN END EL [7453]
SE IF TEST(x,y)=2 THEN PEN 1:LOCATE s(a),z
(a):PRINT CHR$(143)CHR$(143):NEXT
140 IF TEST(x,y)=1 THEN 90                    [832]
150 PLOT x,y:r=0:GOTO 90                     [1348]
```

Listing Misthaufen

```
100 DEG:MODE 1:PEN 3:PAPER 0:FOR i=0 TO 3: [4837]
INK i,i*8:NEXT:BORDER 5
110 INPUT"Geburtsdatum T,M,J: ",s,n,i:INP [3892]
UT"heutiges Datum T,M,J:",t,m,j
120 MODE 1:PRINT TAB(7)"heutiges Datum:";t [3686]
;".";m;".";j:PRINT
130 PEN 1:PRINT"physisch";:PEN 2:PRINT TAB [6622]
(17);"psychisch";:PEN 3:PRINT TAB(34);"gei
stig"
140 GOSUB 210:h=g:m=n:j=i:GOSUB 210:a=h-g- [2268]
s
150 PAPER 1:PRINT:PRINT USING"Geburtstag: [5190]
##. ##. ####      Alter: #####";s;m;j;a+t:PA
PER 0
160 ORIGIN 0,152,10,630,16,288:CLG 0:FOR i [2859]
=1 TO 31:IF i MOD 5=0 THEN y=8 ELSE y=4
170 PLOT i*20-10,-y,3:DRAWR 0,y*2:NEXT i:P [4998]
LOT t*20-10,-136:DRAWR 0,272
180 FOR p=1 TO 3:q=p*5+18:FOR g=1 TO 31:w= [4773]
SIN(360*(a+g)/q)*136
181 IF g=t THEN LOCATE p*16-15,4:PEN p:PRI [3571]
NT USING"###.# %";(w+136)/2.72
190 IF g=1 THEN PLOT 10,w,p ELSE DRAW g*20 [2272]
-10,w
200 NEXT g,p:CALL &BB18:RUN                   [1490]
210 f=m+13:g=j-1:IF m>2 THEN f=f-12:g=g+1  [2445]
220 g=INT(365.25*g)-693975+INT(30.6*f):RET [1960]
URN
```

Listing Biorhythmus

CPC-ANWENDUNG



Autor: Matthias Uphoff

Damit das Schreiben wieder Spaß macht

Der Klassiker:

ConText CPC – bis heute ungeschlagen in der Gruppe der Textverarbeitungen. Dieses Programm besticht vor allem durch seine leichte Anwendungsweise, die selbst unerfahrenen Computerbesitzern den Umgang mit einer Textverarbeitung möglich macht.

ConText CPC – das heißt:

Einfachste Bedienung durch logische Tastaturbelegung; alle Funktionen sind über die Control- und eine definierte Taste zu erreichen.
Funktionen wie EINFÜGEN, FLIESSTEXT, BLOCKFORMATIERUNG und KOPIEREN sind über Tastendruck aufrufbar und werden in einer Statuszeile angezeigt.
Mehrspaltiges Schreiben und Textkopieren erleichtern Ihnen die Korrespondenz.
25 KByte Textspeicher mit insgesamt 5 DIN-A4-Seiten, damit Sie auch lange Briefe problemlos erstellen können.
Voreingestellt für die meisten CENTRONICS-Drucker, durch übersichtliche Tabellenprogrammierung anpaßbar an viele EPSON-kompatible Drucker.
Weiterschreiben während des Druckens, denn 'time is money'.
Darstellung von Sonderschriften wie VERGRÖSSERN und UNTERSTREICHEN, Anzeigen von anderen Schriftarten durch Steuerzeichen.
Auf Diskette / Kassette gespeicherte Textbausteine sind überall im Text plazierbar.
Eingebauter Taschenrechner und Kalender, damit Sie den Überblick behalten.
Dies sind nur einige der Möglichkeiten, die ConText Ihnen als Textverarbeitung bietet.

Der benötigte Hardware-Aufwand ist gering.

Sie brauchen nur einen CPC 464 / 664 oder 6128 und einen Drucker. Alles andere erklärt Ihnen die ausführliche deutsche Bedienungsanleitung, die dem Programm beiliegt.

Und wo gibt's ConText CPC ?

Bei DMV zu bestellen als

3"-Diskette 59,– DM*

Turbo-DATA-CPC

Volldampf in der Dateiverwaltung

Brauchen Sie ein neues Adreßbuch oder Telefonverzeichnis? Müssen Sie Ihre Sammlungen von Briefmarken, Schallplatten, Dias oder anderen Schätzen sortieren? Alles kein Problem, wenn Sie für Ihren CPC eine universelle Dateiverwaltung haben, und die gibt's jetzt beim DMV:

Turbo-DATA-CPC ist die universelle Dateiverwaltung für jede Gelegenheit, mit der Sie problemlos Überblick und Ordnung in alle Ihre Daten bringen können. Dabei helfen Ihnen die vielfältigen Funktionen von Turbo-DATA-CPC:

- Universell durch veränderbare Ein- und Ausgabemasken
- Blitzschnelle Suchfunktionen durch Indexfelder
- Dateikapazität bis maximal 80 verschiedene Felder
- Bis zu 19 Felder gleichzeitig auf dem Bildschirm sichtbar
- Besonders schnelle und umfangreiche Sortierfunktion
- Eigene Formatierroutine mit extrem hoher Kapazität
- Zweiteilung des Bildschirms in Status- und Arbeitsfeld
- Deutscher Zeichensatz und deutsche Tastaturbelegung
- Komplette Druckroutinen, auch für Etikettendruck

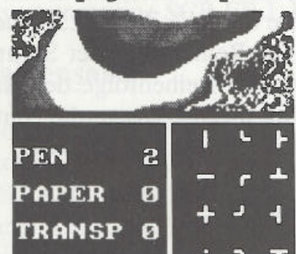
Hardwarevoraussetzung: CPC 464/664/6128
mit einem Diskettenlaufwerk

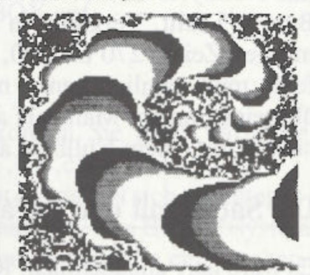
Ganz gleich, was Sie zu archivieren haben, Turbo-DATA-CPC ist in jedem Fall die richtige Wahl.

DM 69,– *

Copyshop

Das universelle Hardcopy-Programm für CPC 464/664/6128
Autor: Matthias Uphoff





Copyshop im Detail:

– Hardcopy in 4 (!) Formaten: DIN A4, DIN A5, 13,5 x 8,5 cm und 21,5 x 13,5 cm – superschnelle Hardcopy-Routine: DIN A4 in ca. 4 Minuten (DMP 4000) – arbeitet in allen drei Modes – Anpaßmenü für JEDEN Epson-kompatiblen Drucker – läuft ebenfalls mit den Seikosha-Druckern GP-500 CPC, GP-550 CPC und GP-1000 CPC – Anpassung an Drucker möglich, die mit 1280 Punkten pro Zeile arbeiten, z.B. CPA-80 GS – Okimate ML 182 – Anpassung kann beim Verlag angefordert werden – Anpassung auch für Drucker möglich, die die Bitbild-Bytes verkehrt herum drucken (das MSB unten statt oben), z.B. NEC P2-Pinwriter. – 32 Farbraster über Menü wählbar – Grafikeditor – komfortable Pulldown-Menüs – Schnelle Fill-Routine – beliebige Ausschnittvergrößerungen – Bildschirm invertieren – selbstrelozierende Hardcopy-Routinen für eigene Programme – neue Save- und Load-Routinen erkennen automatisch Mode und Farbwerte – Freezer – saved auf Tastendruck Screenshots aus laufenden BASIC-Programmen, die anschließend ausgedruckt werden können.

Und die Weltneuheit: **Hardcopy-Simulation auf dem Bildschirm!**
Sie können sich Ihre Hardcopy vor dem endgültigen Ausdruck auf dem Bildschirm ansehen!

3"-Diskette 49,– DM*

SPECIAL OFFERs!

für CPC 464-664-6128, nur auf 3"- Disketten

Original CPC-Software im Paket zu stark herabgesetzten Preisen

DISKSORT-STAR
Leistungsstarke Diskettenverwaltung, die keinem CPC-Benutzer fehlen sollte. DISKSORT-STAR verwaltet, archiviert, katalogisiert, druckt, ... Ihre Diskettensammlung auf einfachste Weise. Neben der reinen Diskettenverwaltung ist unter anderem noch ein kompletter Diskettenmanager enthalten. Auch in puncto Bedienungskomfort ist DISKSORT-STAR kaum zu schlagen.

STAR-MON
Das Entwicklungssystem für Profis
• Assembler • Editor • Disassembler • Monitor • vier Breakpoints • Trace-Funktion • Bankswitch • Memory Dump • Diskettenmonitor • u.v.m.



CREATOR-STAR
Ein Trickfilmdesigner für alle Hobbyregisseure auf dem CPC!
• Sprite-Designer • Laufschrift • Utilities • Kulissendesigner • Sprites mit 4 Unterpositionen • Verbinden von Sprites • Kulissen auch übereinandergelegt • Eigene Programmiersprache mit Editor und Compiler

DM 59,– *

DESIGNER-STAR
Grafikprogramm, mit dem man Bildschirmgrafiken komfortabel erstellen kann. Hilfsmenü auf Tastendruck – kein Joystick oder Maus notwendig.

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

Null Problemo mit RSX

Wie RSX-Programme geschrieben werden

"Nobody is perfect", das weiß jeder. Deshalb haben sich die Entwickler des CPC-BASIC etwas ganz besonderes einfallen lassen: Die Befehlserweiterung mit RSX-Befehlen, mit denen Sie die Befehle Ihres CPC fast beliebig erweitern können. Diese Möglichkeit der Befehlserweiterung ist wirklich einmalig auf der großen Palette der BASIC-Dialekte. Mit diesem Bericht wollen wir Anregungen, Tips und Hintergrundinformationen liefern. Gewisse Kenntnisse in Maschinensprache sollten allerdings schon vorhanden sein.

Sie werden sich wahrscheinlich fragen, was "RSX" eigentlich bedeutet. RSX ist die Abkürzung von 'Resident System Extention' und heißt soviel wie 'feste Systemerweiterung'. Doch was soll das Ganze? Nehmen wir einmal an, Sie hätten ein kleines Assemblerprogramm geschrieben, das den Bildschirm Ihres CPC löscht, das heißt, den Bildschirmspeicher mit Nullbytes füllt (siehe Listing 1). Wenn Sie das Programm mit dem CPC-Assembler aus Heft 6/86 assembliert haben, können Sie es mit CALL &A000 aufrufen. Jedesmal, wenn Sie den Bildschirm löschen wollen, müssen Sie die Routine mit CALL &A000 aufrufen. Das geht ja noch, aber was ist, wenn Sie gleich zehn Assemblerroutinen gleichzeitig im Speicher haben und wegen so vieler Zahlen die Adresse jeder einzelnen Routine gar nicht mehr im Kopf haben? Hier hilft die Befehlserweiterung mit RSX. Damit können Sie jeder einzelnen Routine einen einprägsamen Namen geben. In unserem Beispiel wollen wir der Löschroutine den Namen CLS geben. Sehen Sie sich dazu Listing 2 an; eine Erläuterung wird gleich folgen. Nach der Assemblierung und dem Aufruf der Routine mit CALL &A000 bindet Ihr CPC den Befehl CLS bis zum Ausschalten in das BASIC ein. Wenn Sie alles richtig abgetippt haben, können Sie die Löschroutine mit |CLS aufrufen (das | steht für den sog. RSX-Strich, den Sie erhalten, indem Sie SHIFT und die Taste rechts neben dem 'P' drücken). Der RSX-Strich leitet immer einen RSX-Befehl ein.

Doch zurück zu Listing 2. In Zeile 30 wird die Routine KL LOG EXT mit 'CALL &BCD1' aufgerufen. Diese Betriebssystemroutine bindet einen oder mehrere RSX-Befehle ein. Im HL-Register muß ihr die Adresse von vier Systembytes übergeben werden, die der CPC intern zur Verwaltung der RSX-Befehle benötigt. Im BC-Register muß die Adresse einer Tabelle stehen, deren erste zwei Bytes die Adresse einer zweiten Tabelle angeben müssen. Diese zweite Tabelle muß die Namen Ihrer RSX-Befehle enthalten. Auf die gerade genannten zwei Bytes müssen die Sprungbefehle zu den einzelnen Routinen folgen (Zeile 170). In der Namenstabelle stehen, wie vorher erwähnt, die Namen der RSX-Befehle und zwar in der Reihenfolge, in der die Sprungbefehle stehen. Beim letzten Buchstaben eines Namens muß Bit 7 gesetzt sein (Zeile 190), damit der CPC eventuell mehrere RSX-Befehle unterscheiden kann. Die Namenstabelle wird mit einem Nullbyte abgeschlossen (Zeile 200). Zugegeben, das Ganze ist etwas schwierig, deshalb sollten Sie sich Listing 2 noch einmal ganz genau anschauen.

Wenn Sie Listing 2 mit dem Assembler assembliert haben, können Sie das Programm mit 'CALL &A000' starten. Sie werden sich vielleicht wundern, daß sich der CPC mit einem schlichten 'Ready' meldet und nicht den Bildschirm löscht. Das hat einen wichtigen Grund: der CPC hat soeben nur einen Befehl eingebunden. Geben Sie jetzt aber '|CLS' ein, wird der Bildschirm gelöscht.

Vorsicht, Absturzgefahr

Eines müssen Sie beim Arbeiten mit RSX-Programmen allerdings immer beachten: ein und dieselbe Erweiterung darf nur **einmal** eingebunden werden. Das leuchtet ein, wenn man weiß, daß die Betriebssystemroutine KL LOG EXT alle RSX-Tabellen intern mit Zeigern verbindet. Findet der CPC einen RSX-Befehl in einer Tabelle nicht, so sucht er in einer anderen Tabelle weiter, deren Adresse durch einen Zeiger angegeben wird. Wird eine Erweiterung mehrmals initialisiert, so wird der Zeiger wieder auf die gleiche Tabelle gerichtet. Findet der CPC nun einen RSX-Befehl innerhalb dieser Tabelle nicht, so kommt er unweigerlich in eine Endlosschleife, was natürlich einen Absturz des Systems zur Folge hat. Also noch einmal: dieselbe Erweiterung kein zweites Mal aufrufen, und schon kann (fast) nichts schiefgehen. Bisher haben wir nur Programme besprochen, in denen nur ein RSX-Befehl eingebunden werden mußte. Sehen Sie sich bitte Listing 3 an. Es soll Ihnen anschaulich machen, wie gleich mehrere RSX-Befehle eingebunden werden können. Wenn Sie das Listing mit dem CPC-Assembler assembliert und mit 'CALL &A000' gestartet haben, stehen Ihnen zwei neue RSX-Befehle zur Verfügung: |BLACK, der den Bildschirm mit Nullbytes und |WHITE, der den Bildschirm mit dem Byte 255 füllt. Vergleichen Sie bitte die Zeilen 250 und 260 mit den Zeilen 270 bis 300. Die Reihenfolge der Sprungbefehle muß nämlich genau mit der Reihenfolge der Namen übereinstimmen. Man darf aber nie vergessen, die Namenstabelle mit einem Nullbyte abzuschließen (Zeile 310).

Die Sache mit den Parametern

Es wäre naiv anzunehmen, RSX diene nur dazu, Maschinensprachprogrammen einen Namen zu geben. Nein, Sie können auch vom BASIC aus Parameter an das Maschinensprachprogramm übergeben. Die Parameterübergabe ist übrigens die gleiche wie bei dem CALL-Befehl und sieht folgendermaßen aus: **|NAME, Parameter 1, Parameter 2, Parameter 3, Parameter 4,...**

Sie können bis zu 32 Parameter von BASIC an das Maschinensprachprogramm übergeben. Wie können Sie nun die vom BASIC übergebenen Parameter ins Maschinensprachprogramm übernehmen? Hier helfen Ihnen die Register und Flags. Nach der Eingabe eines RSX-Befehls stehen in den Registern und Flags folgende Werte:

Register	*Bedeutung*
A	gibt Anzahl der übergebenen Parameter an
IX	Adresse des letzten Parameters. IX+2 enthält folglich den vorletzten, IX+4 den vorvorletzten Parameter usw.

Register	*Bedeutung*
DE	enthält den letzten übergebenen Parameter
B	enthält 32-Anzahl der Parameter

Flags	*Bedeutung*
Zero-Flag	keine Parameter: Zero-Flag gesetzt, sonst nicht gesetzt

Listing 4 soll Ihnen das soeben Beschriebene anschaulich zeigen. Es installiert nach dem Assemblieren und Starten mit 'CALL &A000' einen neuen RSX-Befehl mit dem Format: |FILLBOX, linke, rechte, obere, untere physikalische Koordinate, Füllbyte
Der Begriff 'physikalische Koordinate' wird wohl manchen irritieren, er bedeutet aber nichts anderes als 'Textkoordinate' mit dem Ursprung 0,0 links oben. |FILLBOX, 0, 19, 0, 4, 255 füllt zum Beispiel einen Bereich links oben am Bildschirm mit der Breite von 20 und Länge von 5 Textkoordinaten mit dem Byte 255. Doch zurück zum Listing: die Zeilen 80 bis 110 und 210 bis 260 installieren den RSX-Befehl. Die Zeile 120 prüft, ob auch fünf Parameter übergeben wurden. Wenn das nicht der Fall ist, wird das Programm beendet. Diese Überprüfung sollten Sie übrigens bei jedem Maschinensprachprogramm, das vom BASIC Parameter übernimmt, einbauen. Fehleingaben werden seltener und das Programm wird anwenderfreundlicher. In den Zeilen 140 bis 180 übernimmt das Maschinensprachprogramm die benötigten Werte und transferiert sie in die entsprechenden Register. Die Highbytes werden hierbei nicht berücksichtigt, da ja kein Wert größer als 255 ist. Hier sollten Sie beachten, daß der CPC die mit einem RSX-Befehl übergebenen Parameter in eine Tabelle überträgt.
Das IX-Register gibt dann die letzte Adresse an. IX gibt damit das Lowbyte, IX+1 das Highbyte des letzten Parameters, IX+2 das Lowbyte, IX+3 das Highbyte des vorletzten Parameters usw. an. Die Zeile 190 ruft schließlich die Betriebssystemroutine SCR FILL BOX auf.

Variablensalat

Bei dem vorigen RSX-Befehl können Sie auch Variablen übergeben, z.B.: |FILLBOX,a%,b%,c%,d%,e%.
Das '%' hinter jeder Variable gibt an, daß es sich um Integer- (das heißt Ganzzahl-) Variablen handelt. Das Betriebssystem übergibt der Maschinensprachroutine dann die entsprechenden Werte. Es gibt allerdings noch einen zweiten Fall: Sie können Variablen übergeben, deren Adresse der Maschinensprachroutine übergeben werden. Sie können dann die Variablen vor der Rückkehr zum BASIC manipulieren, so daß Sie Werte vom Maschinensprachprogramm an das BASIC zurückgeben.
Solche Variablen müssen vorne mit einem sogenannten 'Klammeraffen' gekennzeichnet werden. Diesen 'Klammeraffen' erreichen Sie durch Druck auf die Taste rechts neben dem 'P'. Dem Maschinenspracheprogramm wird dann nicht wie vorher der Wert, sondern eine sogenannte Deskriptoradresse der Variablen übermittelt. Diese Deskriptoradresse gibt bei einer Integervariablen (wir wollen uns auf eine solche beschränken) die Adresse des Wertes an, den die Variable repräsentiert. Nach dem Motto "ein Beispiel sagt mehr aus als tausend Worte" wollen wir das soeben Gesagte zunächst einmal im BASIC ausprobieren. Geben Sie doch einmal folgendes ein: *a%=300:PRINT* @a% Der BASIC-Interpreter wird nun irgendeine Adresse auf dem Bildschirm ausgeben. Jetzt geben Sie bitte folgendes ein: *PRINT PEEK(*@a%),PEEK(@a%+1) Jetzt wird der CPC zuerst eine 44 und dann eine 1 ausgeben. Die 44 gibt das Lowbyte, die 1 das Highbyte des Wertes an. Nach der altbekannten Formel Highbyte mal 256 plus Lowbyte können wir den Variableninhalt berechnen, also: 1 mal 256 plus 44. Das ergibt 300.

Adresse	Name	Funktion	Input	Output
&BCD1	KL LOG EXT	bindet einen RSX-Befehl ein	HL: Adr. von 4 Bytes BC: Adr. von Tabelle	
&BCD4	KL FIND COMMAND	sucht einen RSX-Befehl	HL: Adr. von Name	HL: Adresse C: ROM-Nr.
&BC44	SCR FILL BOX	füllt einen rechteckigen Bereich im Bildschirm	A:codierte Ink H:linke Spalte D:rechte " L:obere Reihe E:untere "	
&BB60	TXT READ CHAR	liest Zeichen vom Bildschirm		A: Zeichen
&BB75	TXT SET CURSOR	setzt Text-Cursor	H:Spalte L:Zeile	
&BB5A	TXT OUTPUT	gibt ein Zeichen auf dem Bildschirm aus	A:Zeichen	

Tabelle der Systemroutinen in Zusammenhang mit RSX-Befehlen

Wir können den Variableninhalt aber auch manipulieren. Sie wollen eine 10 in der Variablen "a%"? Nichts einfacher als das: Das Lowbyte von 10 ist 10, das Highbyte 0. Mit *POKE* @a%,10:POKE @a%+1,0 schreiben wir in die Variable a% den Wert 10. Mit *PRINT a%* können Sie das überprüfen.

Kommen wir aber wieder zurück zur RSX-Programmierung! Nehmen wir einmal an, Sie wollen einen RSX-Befehl schreiben, der vom Bildschirm das Zeichen einer bestimmten Cursor-Position liest (auf den CPCs 664/6128 gibt es in BASIC mit COPYCHR$ schon einen entsprechenden Befehl). Der RSX-Befehl sollte folgendes Format haben: |READ, x, y, @wert%. Die Variablen x und y geben die entsprechenden Textkoordinaten an und die Variable wert% gibt nach der Rückkehr zum BASIC den ASCII-Wert des Zeichens an, das sich an der betreffenden Stelle befindet. Für die Programmierung des gewünschten Befehls benötigen wir zwei Betriebssystemroutinen: TXT SET CURSOR, die den Text-Cursor an eine bestimmte Stelle plaziert (analog zum BASIC-Befehl LOCATE) und TXT RD CHAR, die von der aktuellen Cursor-Position ein Zeichen liest (analog zum BASIC-Befehl COPYCHR$ bei den CPC 664/6128). Listing 5 zeigt die Programmierung des |READ-Befehls. Die Zeilen 50 bis 70 und 230 bis 280 installieren den Befehl. Zeile 90 prüft, ob drei Befehle übergeben wurden. Die Zeilen 110 und 120 übergeben die Deskriptoradresse an das HL-Register. Zeile 130 legt das HL-Register auf dem Stapel ab. Die Zeilen 140 bis 170 holen die Koordinaten, setzen den Text-Cursor und lesen das Zeichen vom Bildschirm. Zeile 180 holt die Deskriptoradresse wieder vom Stapel und speichert darin den ASCII-Wert des Zeichens. In Zeile 210 wird in das Highbyte der Deskriptoradresse eine Null geladen, weil der ASCII-Wert nie über 255 hinausgeht. Zeile 220 veranlaßt den CPC schließlich zur Rückkehr ins BASIC.

Nach der Assemblierung und dem Start mit 'CALL &A000' steht Ihnen der |READ-Befehl zur Verfügung. Vor Aufruf des Befehls muß die Variable wert% aber definiert sein, sonst gibt der CPC unweigerlich ein 'Improper Argument' aus. Beispiel: *wert%=0:*|READ,1,1,@wert%:PRINT wert% Der CPC gibt nun den ASCII-Wert des Zeichens aus, das sich an der Koordinate 1,1 befindet. Wenn der CPC eine Null ausgibt, konnte er das Zeichen nicht identifizieren.

RSX und Strings

Manche Leser werden sich nun fragen, wie man Strings (Zeichenketten) als Parameter übergibt. Auch hier benötigen wir wieder unseren 'Klammeraffen' (auf dem CPC 664/6128 gibt es eine weitere Möglichkeit der Stringübergabe, die wir hier aber aus Kompatibilitätsgründen außer acht lassen). Wie bei den Integervariablen wird bei Strings dem Maschinensprachprogramm eine Deskriptoradresse übergeben. Diese unterscheidet sich aber erheblich von der vorigen Integer-Deskriptoradresse. Zuerst wird nämlich die Länge des Strings angegeben.

Darauf folgen das Low- und Highbyte der Stringadresse. Hierzu muß gesagt werden, daß sich diese Adresse durch die sogenannte 'Garbage Collection' laufend ändern kann. Auch hierzu sollten wir uns ein Beispielprogramm anschauen, nämlich Listing 6. Es initialisiert einen RSX-Befehl, der folgendes Format besitzt: |PRINT,@a$. Er soll einen String, hier a$, auf dem Bildschirm ausgeben (analog zum BASIC-Befehl PRINT). Nehmen wir das Listing einmal unter die Lupe: Die Zeilen 50 bis 70 und 250 bis 300 initialisieren den |PRINT-Befehl. Zeile 90 prüft, ob ein Parameter übergeben wurde.

Ist das der Fall, so wird die Deskriptoradresse ins HL-Register übertragen. Die folgenden Zeilen laden das B-Register mit der Länge und das DE-Register mit der Adresse des Strings. Danach wird innerhalb einer Schleife der String mit Hilfe der Betriebssystemroutine TXT OUTPUT auf dem Bildschirm ausgegeben. Mit 'RET' kommt der CPC dann wieder in den Direktmodus zurück.

Auf den Spuren Sherlock Holmes'

Bisher haben wir mit der Betriebssystemroutine KL LOG EXT RSX-Befehle nur eingebunden. Mit einer anderen Betriebssystemroutine, nämlich KL FIND COMMAND, können wir bereits vorhandene RSX-Befehle suchen. Listing 7 macht das anhand des RSX-Befehls |DIR anschaulich, der nur bei den CPCs mit Diskettenlaufwerk vorhanden ist. Der Routine KL FIND COMMAND muß im HL-Register die Adresse des Namens übergeben werden (Zeile 40). Bei dem letzten Buchstaben muß aber Bit 7 gesetzt sein (Zeile 160 und 170). Zeile 50 ruft KL FIND COMMAND mit 'CALL &BCD4' auf. Wenn das Carry-Flag gesetzt ist, kann der CPC den RSX-Befehl finden. Im HL-Register befindet sich dann die Anfangsadresse und im C-Register die ROM-Nummer. Auf diese Betriebssystemroutine soll aber nicht weiter eingegangen werden, weil sie sehr selten benutzt wird.

Abschließend finden Sie noch eine Liste für weiterführende Literatur, die sich mit der RSX-Programmierung beschäftigt, und eine Zusammenstellung aller Betriebssystemroutinen, die wir in den Beispielprogrammen benutzt haben. Nach dem Motto "Probieren geht über Studieren" wünschen wir Ihnen noch viel Spaß und wenig Abstürze bei der RSX-Programmierung!

Weiterführende Literatur

Dullin/Strassenburg, Das Maschinensprachebuch zum CPC, Verlag Data Becker

K.Kremer, Profi RSX, PC Schneider International 6/87-1/88

M.Uphoff, Die Assemblerecke, PC Amstrad International ab 10/88

(Markus Felder/cd)

für 464-664-6128

```
10 ';****** Listing 1 ******
20 ';
30 'ORG &A000           ;Beginn des Maschinencodeprogra
mms
40 'LD HL,&C000         ;Anfang des Bildschirmspeichers
 in HL laden
50 'LOOP XOR A          ;A mit sich selbst xorieren, al
so mit Null laden
60 'LD (HL),A           ;A in HL speichern
70 'INC HL              ;HL erhoehen
80 'LD A,H              ;A mit H laden
90 'CP 0                ;HL groesser als 65535 ?
100 'JR NZ,LOOP          ;Wenn Nein, dann wiederhole
110 'RET                 ;zurueck zum Basic
```

```
10 ';****** Listing 2 ******
20 ';
30 'ORG &A000            ;Beginn des Maschinencodeprogr
amms
40 'LD HL,SYSTEM         ;Adresse der 4 Systembytes lad
en
50 'LD BC,TABELL         ;Adresse der 1.Tabelle laden
60 'CALL &BCD1           ;KL LOG EXT aufrufen
70 'RET                  ;zurueck ins Basic
80 'CLS LD HL,&C000      ;Beginn der Loeschroutine
90 'LOOP XOR A           ;vgl. Listing 1...
100 'LD (HL),A
110 'INC HL
120 'LD A,H
130 'CP 0
140 'JR NZ,LOOP
150 'RET
160 'TABELL DW NAMTAB;Anfangsadresse der Namenstab
elle
170 'JP CLS              ;Sprungbefehl auf Loeschrouti
ne
180 'NAMTAB DM "CL"      ;Namenstabelle
190 'DB 211              ;beim letzten Buchstaben Bit
7 setzen: 211=ASC("S")+128
200 'DB 0                ;Ende der Namenstabelle
210 'SYSTEM DS 4         ;4 Bytes reservieren
```

```
10 ';****** Listing 3 ******
20 ';
30 'ORG &A000            ;Beginn des Maschinencodeprogr
amms
40 'LD HL,SYSTEM         ;Adresse der 4 Systembytes lad
en
50 'LD BC,TABELL         ;Adresse der 1.Tabelle laden
60 'CALL &BCD1           ;KL LOG EXT aufrufen
70 'RET                  ;zurueck ins Basic
80 'BLACK LD HL,&C000;Beginn der BLACK-Routine
90 'LOOP1 XOR A          ;A mit 0 laden
100 'LD (HL),A           ;A in HL speichern
110 'INC HL              ;HL erhoehen
120 'LD A,H              ;A mit H laden
130 'CP 0                ;HL groesser als 65535?
140 'JR NZ,LOOP1         ;wenn nicht, dann wiederhole
150 'RET                 ;zurueck zum Basic
160 'WHITE LD HL,&C000;Beginn der WHITE-Routine
170 'LOOP2 LD A,255 ;A mit 255(Bitmuster:&X1111111
1) laden
180 'LD (HL),A           ;A in HL speichern
190 'INC HL              ;HL erhoehen
200 'LD A,H              ;A mit H laden
210 'CP 0                ;HL groesser als 65535?
220 'JR NZ,LOOP2         ;wenn nicht, dann wiederhole
230 'RET                 ;zurueck zum Basic
```

Listing RSX

```
240 'TABELL DW NAMTAB;Anfangsadresse der Namenstab
elle
250 'JP BLACK        ;Sprungbefehl auf 1.RSX-Befeh
l
260 'JP WHITE        ;Sprungbefehl auf 2.RSX-Befeh
l
270 'NAMTAB DM "BLAC";Namenstabelle: 1.RSX-Befehl
280 'DB 203          ;203=ASC("K")+128
290 'DM "WHIT"       ;2.RSX-Befehl
300 'DB 197          ;197=ASC("E")+128
310 'DB 0            ;Ende der Namenstabelle
320 'SYSTEM DS 4     ;4 Bytes reservieren
```

```
10 ';****** Listing 4 ******
20 ';
30 ';Befehlsform: |FILLBOX,linke,rechte,obere,unte
re physikalische Koordinate des Bereichs,Fuellbyte
40 ';verwendete Betriebssystemroutine: &BC44 fuel
lt einen rechteckigen Zeichenbereich des Bildschir
ms mit einer codierten Ink
50 ';Input: A=codierte Ink,H=linke,D=rechte physik
alische Spalte,L=obere,E=untere physikalische Reih
e
60 ';ACHTUNG: Fehleingaben koennen zum Absturz fue
hren!!!
70 'ORG &A000         ;Beginn des Maschinenspracher
ogramms
80 'LD HL,SYSTEM      ;Adresse der 4 Systembytes lad
en
90 'LD BC,TABELL      ;Adresse der 1.Tabelle laden
100 'CALL &BCD1       ;KL LOG EXT aufrufen
110 'RET              ;und zurueck ins Basic
120 'BOX CP 5         ;Beginn der Routine: 5 Parame
ter???
130 'RET NZ           ;wenn NEIN: zurueck ins Basic
140 'LD A,(IX)        ;Lowbyte des Fuellbytes in de
n Akku laden
150 'LD E,(IX+2)      ;Lowbyte des vorletzten Param
eters ins E-Register laden
160 'LD L,(IX+4)      ;Lowbyte des 3.Parameters ins
 L-Register laden
170 'LD D,(IX+6)      ;Lowbyte des 2.Parameters ins
 D-Register laden
180 'LD H,(IX+8)      ;Lowbyte des 1.Parameters ins
 H-Register laden
190 'CALL &BC44       ;SCR FILL BOX aufrufen
200 'RET              ;und zurueck ins Basic
210 'TABELL DW NAMTAB ;Anfangsadresse der Namensta
belle
220 'JP BOX           ;Sprungbefehl auf Fuellroutin
e
230 'NAMTAB DM "FILLBO" ;Namenstabelle
240 'DB 216           ;216=ASC("X")+128 -->beim let
zten Buchstaben Bit 7 setzen
250 'DB 0             ;Namenstabelle abschliessen
260 'SYSTEM DS 4      ;4 Bytes fuer Betriebssystem
reservieren
```

```
10 ';****** Listing 5 ******
20 ';
30 ';Format des Befehls: |READ,x,y,@wert%
40 'ORG &A000         ;Beginn des Maschinencodeprogr
amms
50 'LD HL,SYSTEM      ;Adresse der 4 Systembytes lad
en
60 'LD BC,TABELL      ;Adresse der 1.Tabelle laden
70 'CALL &BCD1        ;KL LOG EXT aufrufen
80 'RET               ;zurueck ins Basic
90 'READ CP 3         ;Beginn der Routine: 3 Paramet
er???
100 'RET NZ           ;wenn NEIN: zurueck ins Basic
110 'LD L,(IX)        ;Lowbyte der Deskriptoradress
e laden
120 'LD H,(IX+1)      ;Highbyte "
 "
130 'PUSH HL          ;und retten
140 'LD L,(IX+2)      ;Y-Koordinate holen
150 'LD H,(IX+4)      ;X-Koordinate holen
160 'CALL &BB75       ;TXT SET CURSOR aufrufen
170 'CALL &BB60       ;TXT READ CHAR aufrufen
```

Listing RSX

```
180 'POP HL           ;Deskriptoradresse wieder vom
 Stapel holen
190 'LD (HL),A        ;ASCII-Wert in die Variable l
aden
200 'INC HL           ;Deskriptoradresse erhoehen
210 'LD (HL),0        ;und ins Highbyte eine 0 lade
n
220 'RET              ;zurueck ins Basic
230 'TABELL DW NAMTAB ;Anfangsadresse der Namensta
belle
240 'JP READ          ;Sprungbefehl auf READ-Routin
e
250 'NAMTAB DM "REA"  ;Namenstabelle
260 'DB 196           ;196=ASC("D")+128 --> Bit 7 s
etzen
270 'DB 0             ;Namenstabelle abschliessen
280 'SYSTEM DS 4      ;4 Bytes fuer Betriebssystem
reservieren
```

```
10 ';****** Listing 6 ******
20 ';
30 ';Format des Befehls: |PRINT,@a$
40 'ORG &A000         ;Beginn des Maschinencodeprogr
amms
50 'LD HL,SYSTEM      ;Adresse der 4 Systembytes lad
en
60 'LD BC,TABELL      ;Adresse der 1.Tabelle laden
70 'CALL &BCD1        ;KL LOG EXT aufrufen
80 'RET               ;zurueck ins Basic
90 'PRINT CP 1        ;Beginn der Routine: 1 Paramet
er???
100 'RET NZ           ;wenn NEIN: zurueck ins Basic
110 'LD L,(IX)        ;Lowbyte der Deskriptoradress
e laden
120 'LD H,(IX+1)      ;Highbyte "                "
130 'LD B,(HL)        ;B mit Laenge des Strings lad
en
140 'INC HL           ;HL erhoehen
150 'LD A,(HL)        ;A mit Inhalt von HL laden
160 'LD E,A           ;und ins E-Register uebertrag
en
170 'INC HL           ;HL erhoehen
180 'LD A,(HL)        ;A mit Inhalt von HL laden
190 'LD D,A           ;und ins D-Register uebertrag
en
200 'LOOP LD A,(DE)   ;A mit Inhalt von DE laden
210 'CALL &BB5A       ;TXT OUTPUT aufrufen
220 'INC DE           ;DE erhoehen
230 'DJNZ LOOP        ;zu LOOP springen bis B=0 ist
240 'RET              ;zurueck ins Basic
250 'TABELL DW NAMTAB ;Anfang der Namenstabelle
260 'JP PRINT         ;Sprungbefehl auf PRINT-Routi
ne
270 'NAMTAB DM "PRIN" ;Namenstabelle
280 'DB 212           ;212=ASC("T")+128 --> Bit 7 s
etzen
290 'DB 0             ;Namenstabelle abschliessen
300 'SYSTEM DS 4      ;4 Bytes fuer Betriebssystem
reservieren
```

```
10 ';****** Listing 7 ******
20 ';
30 'ORG &A000         ;Beginn des Maschinencodeprogr
amms
40 'LD HL,NAME        ;HL mit Name des RSX-Befehls l
aden
50 'CALL &BCD4        ;KL FIND COMMAND aufrufen
60 'JR C,WEITER       ;Gefunden: dann weiter
70 'LD A,7            ;sonst Piepston
80 'CALL &BB5A        ;ausgeben
90 'RET               ;und zum Basic zurueckkehren
100 'WEITER LD (TAB),HL ;Adresse abspeichern
110 'LD A,C           ;A mit C laden
120 'LD (ROM),A       ;ROM-Nummer abspeichern
130 'RET
140 'TAB DS 2         ;2 Bytes reservieren
150 'ROM DS 1         ;1 Byte reservieren
160 'NAME DM "DI"     ;Namen
170 'DB 210           ;210=ASC("R")+128 -->bei letz
tem Buchstaben Bit 7 setzen
```

Listing RSX

Verborgene Geheimnisse

Wie Maschinenprogramme hinter BASIC versteckt werden können

Wir möchten Ihnen anhand eines kleinen Grundlagenartikels zeigen, wie man MC-Files durch Anhängen an BASIC lädt, und welche Vorteile sich dadurch für den Anwender bieten.

Die ganze Methode basiert auf einer Eigenheit des BASIC-Interpreters, der mit der Programmabarbeitung aufhört, sobald er vier Nullbytes findet. Dadurch können beliebige Daten an BASIC angehängt werden (hier: MC, aber auch zum Beispiel Texte, die nach derselben Methode angehängt werden, sind möglich). Da diese Daten wie BASIC behandelt werden, muß MC hinter BASIC nicht geschützt werden, und das BASIC-Programm darf nach dem Anhängen nicht mehr verändert werden, um den MC nicht zu verschieben.

Anleitung für das manuelle Anhängen von MC-Programmen an BASIC

1a. Wenn Sie ein MC-Programm an BASIC anhängen wollen, das die Kontrolle übernimmt (Vordergrundprogramm), schreiben Sie ein beliebiges, kurzes BASIC-Programm, in dem als letzter Befehl ein *CALL &FFFF* abgearbeitet wird. (Bem: Die Zahl &FFFF bzw. &FF dient im folgenden als Platzhalter und wird später durch den korrekten Wert ersetzt).
1b. Wenn Sie aber ein Unterprogramm in MC anhängen wollen (Hintergrundprogramm), ersetzen Sie die LOAD-Anweisung durch den Befehl *CALL &FFFF*.
Ist das BASIC- oder MC-Programm sehr lang, schreiben Sie noch: *POKE &AE83, &FF:POKE &AE84,&FF* (bei CPC 664/6128: &AE66 u. &AE67).
Anm: Folgende Befehlsfolge macht das Programm auf allen CPCs lauffähig:



IF PEEK(BBFA)=54 THEN POKE &AE83,&FF:POKE &AE84,&FF ELSE POKE &AE66,&FF:POKE &AE67, &FF.
2. Finden Sie jetzt die Endadresse des BASIC-Programms heraus: Diese steht in den Speicherstellen &AE83 und &AE84 (664/ 6128: &AE66 u. &AE67). Wenn Sie diese Adresse um Eins erhöhen, erhalten sie eine Adresse, die die Basisadresse für MC-Code ist.
3. Tragen Sie die Basisadresse in den CALL-Befehl ein. Haben Sie oben genannte POKE-Befehle verwendet (diese dienen dazu, daß die alte Länge des BASIC-Programms ohne angehängten MC wiederhergestellt wird), tragen Sie das Lowbyte der Basisadresse-1 per POKE in &AE83 (664/6128: &AE66) und das Highbyte in &AE84 (664/6128: &AE67) ein. Speichern Sie das Programm jetzt mit dem SAVE-Befehl wie ein normales BASIC-Programm ab.
4. Finden Sie die Länge des MC-Programms heraus (mit einem Monitor o.ä.).
5. Wenn der anzuhängende MC nicht ortsunabhängig ist, schreiben Sie eine Verschieberoutine vor das MC-Programm, die von der Basisadresse lauffähig ist, und welche den MC an seine ursprünglich richtige Adresse verschiebt. Bei 1a. beenden Sie diese Routine mit einem *JP* an die MC-Startadresse, sonst mit einem *RET*.
Speichern Sie das MC-Programm und die Routine zusammen ab.
6. Laden Sie das BASIC-Programm wieder und reservieren Sie mit folgender Befehlsfolge den Speicherplatz:
OPENOUT"D":MEMORY basisadresse-1:CLEAR LOAD"mcprogramm",basisadresse.
Jetzt addieren Sie zur Basisadresse die Länge des MC-Programms (incl. Schieberoutine) und POKEn den Wert an die Adressen &AE83 u. &AE84 (664/6128: &AE66 u. &AE67); das Lowbyte an die niedrigere, das Highbyte an die höhere Adresse. Jetzt können Sie BASIC und MC zusammen mit dem SAVE-Befehl abspeichern. Das Programm kann jetzt wie ein normales BASIC-Programm benutzt werden. Allerdings darf es nicht mehr in der Länge verändert werden, da dies zu einer Verschiebung des MC und damit wahrscheinlich zum Absturz führen würde.
Wenn Sie ein MC-Vordergrundprogramm anhängen wollen und mehr als nur einen CALL ins MC-Programm im BASIC-Teil brauchen, schreiben Sie ein entsprechendes BASIC-Programm und beenden es mit einem *CALL &FFFF* in der letzten Zeile. Speichern Sie dieses Programm ab. Wenn Sie an ein BASIC-Vordergrundprogramm ein MC-Unterprogramm anhängen wollen, hängen sie eine Zeile mit *END:CALL &FFFF* an und befolgen die Anweisungen unter 1b.
Starten Sie MCHINTER.BAS. Haben Sie ein BASIC-Programm geschrieben, geben Sie dessen Namen ein (bei Kassette max. 16 Zeichen, bei Diskette max. acht Zeichen und drei Zeichen Extension, die durch einen Punkt abgetrennt sind). Wenn Sie kein BASIC-Programm geschrieben haben, drücken Sie einfach RETURN. Geben Sie dann den Namen des anzuhängenden MC-Programms ein und zuletzt den Namen, unter dem das zusammengefügte Programm gespeichert werden soll. Anschließend fragt MCHINTERBAS, ob das MC-Programm vor Aufruf verschoben werden soll. Wenn ja, müssen Sie die (S)tartadresse ("S" drücken) , die in einem normalen BASIC-Ladeprogramm die CALL-Adresse ist, und eventuell die (Z)ieladresse eingeben. Die Zieladresse müssen Sie nur eingeben, wenn im BASIC-Lader hinter dem LOAD eine Adresse steht. Steht also im Programm *LOAD"name",5000:CALL &6000,* müssen Sie als **Startadresse 6000** und **als Zieladresse 5000** eingeben. Wenn Sie ein automatisch startendes MC-Programm anhängen wollen, brauchen Sie (K)eine Adresse eingeben.
Haben Sie alle Eingaben bestätigt, werden Sie aufgefordert, die jeweils zu ladenden Programme einzulegen: zuerst das BASIC-Programm (nur wenn Sie einen Namen eingegeben haben), anschließend das MC-Programm. Danach werden Sie gebeten, einen Datenträger mit Platz für das zusammengefügte Programm einzulegen.
Wurde das zusammengefügte Programm abgespeichert, fragt MCHINTERBAS, ob Sie einen Neustart wollen. Wenn ja, müssen Sie die Kassette/Diskette mit MCHINTERBAS einlegen, sonst erfolgt ein Reset. Haben Sie ein BASIC-(Vordergrund-) Programm und ein MC-(Hintergrund-) Unterprogramm verbunden, müssen Sie noch die unter Punkt 3 aufgeführten Anweisungen befolgen.
Die Basisadresse ist die Adresse im CALL am Ende des Programms. Außerdem müssen Sie noch an die Basisadresse +11 ein &C9 POKEn.
Und nun viel Spaß beim Experimentieren!

(Ulrich Denker/cd)

JAHRESINHALTSVERZEICHNIS PC AMSTRAD INTERNATIONAL 1989

BERICHTE:

ASSEMBLER:

PROGRAMME:

TIPS & TRICKS:

SOFTWARE REVIEWS:

ABENTEUER:

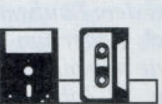
PCW:

PC:

HARDWARE:

(ke)

für 464-664-6128



```
10 '>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>     [1449]
20 '>                                  >     [275]
30 '>      MCHB.DAT  :  DATALADER      >     [412]
40 '>           von  :  MCHB.BIN       <     [1225]
50 '<                                  <     [289]
60 '<         By  Ulrich Denker        <     [622]
70 '<                                  <     [289]
80 '<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     [2058]
90 '<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     [2058]
100 DATA 21,87,A1,11,70,01,01,0D,00,ED,43, [4538]
35,A2,ED,B0,ED,53,33,A2,18,2A,21,07F5
110 DATA 37,A2,CD,2C,A1,B7,C2,37,A1,DD,21, [3124]
70,01,DD,E5,DD,09,DD,22,33,A2,ED,1391
120 DATA 43,35,A2,E1,CD,83,BC,CD,7A,BC,2A, [5077]
33,A2,DD,75,FB,DD,74,FC,21,49,A2,2040
130 DATA CD,2C,A1,FE,02,C2,37,A1,3A,6D,A2, [3950]
B7,C4,A5,A0,2A,35,A2,09,22,35,A2,2A80
140 DATA 2A,33,A2,CD,83,BC,CD,7A,BC,21,5B, [5492]
A2,CD,57,A1,11,00,C0,CD,8C,BC,D2,3629
150 DATA 37,A1,21,70,01,ED,5B,35,A2,01,00, [4219]
00,AF,CD,98,BC,CD,8F,BC,01,F8,A1,4035
160 DATA CD,70,A1,CD,06,BB,E6,DF,FE,4E,CC, [4125]
00,00,21,25,A2,CD,2C,A1,B7,C2,37,4BB0
170 DATA A1,21,70,01,CD,83,BC,CD,7A,BC,C9, [3494]
C5,DD,2A,33,A2,FE,01,38,13,ED,53,56E6
180 DATA 6E,A2,FE,03,20,0B,11,1A,00,19,5E, [3690]
23,56,ED,53,70,A2,21,79,A1,ED,5B,5F12
190 DATA 33,A2,01,0E,00,ED,B0,ED,53,33,A2, [3087]
2A,6E,A2,EB,AF,ED,52,30,12,2A,33,685A
200 DATA A2,C1,09,2B,EB,09,2B,EB,DD,36,0A, [3107]
B8,3E,0E,18,20,ED,5B,33,A2,D5,DD,7223
210 DATA 73,01,DD,72,02,21,79,A1,01,0E,00, [5454]
ED,B0,ED,53,33,A2,DD,E1,C1,2A,6E,7BFB
220 DATA A2,EB,3E,1C,DD,75,01,DD,74,02,DD, [2957]
73,04,DD,72,05,DD,71,07,DD,70,08,85DA
230 DATA 2A,70,A2,DD,75,0C,DD,74,0D,03,3D, [2502]
20,FC,C9,CD,57,A1,11,00,C0,CD,77,8FD1
240 DATA BC,D8,F1,CD,7D,BC,CD,92,BC,01,C0, [4816]
A1,CD,70,A1,CD,06,BB,E6,DF,FE,4E,9E56
250 DATA CA,91,A0,3A,72,A2,B7,C2,00,A0,C3, [3593]
15,A0,01,94,A1,CD,70,A1,7E,23,F5,A9DA
260 DATA 44,4D,CD,70,A1,01,AD,A1,CD,70,A1, [4124]
CD,06,BB,C1,C9,0A,B7,C8,CD,5A,BB,B5F9
270 DATA 03,18,F7,21,00,00,11,C7,B0,01,0E, [4279]
00,ED,B0,C3,C7,B0,0A,00,01,00,83,BD28
280 DATA 20,1C,7D,01,00,00,00,00,0C,0F,03, [4391]
1F,05,07,42,49,54,54,45,20,45,49,C051
290 DATA 4E,4C,45,47,45,4E,3A,20,0F,01,00, [2840]
0F,02,1F,0E,09,44,61,6E,6E,20,54,C4B0
300 DATA 61,73,74,65,20,21,21,00,0D,0A,18, [2827]
20,21,20,46,45,48,4C,45,52,20,21,C946
310 DATA 20,0D,18,0A,0A,4E,65,75,65,72,20, [3447]
28,56,29,65,72,73,75,63,68,20,6F,CF7E
320 DATA 64,65,72,20,28,4E,29,65,75,73,74, [4078]
61,72,74,20,3F,0A,0A,0A,00,0F,01,D50D
330 DATA 18,0C,1F,0A,06,50,72,6F,67,72,61, [4182]
6D,6D,20,62,65,65,6E,64,65,74,2E,DBCA
340 DATA 1F,12,08,4E,65,75,73,74,61,72,74, [3721]
20,3F,20,28,4A,2F,4E,29,18,00,0C,E114
350 DATA 4D,43,48,49,4E,54,45,52,2E,42,41, [3804]
53,00,00,00,00,00,00,00,00,00,00,E472
360 MEMORY &9FFF:zeile=100                 [961]
370 FOR offs=1 TO 572                      [587]
380 READ a$:a=VAL("&"+a$):POKE offs+&9FFF, [2709]
a:pr=pr+a
390 IF pr>65535 THEN pr=pr-65535           [798]
400 IF offs MOD 22<>0 THEN GOTO 440        [2008]
410 READ ps$:ps=VAL("&"+ps$):IF ps<0 THEN  [2766]
ps=ps+65536
420 IF pr<>ps THEN PRINT"Pruefsummenfehler [5238]
 in Zeile";zeile:STOP
430 zeile=zeile+10                         [568]
440 NEXT offs                              [835]
450 SAVE"MCHB.BIN",b,&A000,&233:END        [1442]
```

```
10 '>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>     [1449]
20 '>                                  >     [275]
30 '>          MC-HINTER-BAS           <     [1149]
40 '<                                  <     [289]
```

Listing VERSTECK

```
50 '<        By   Ulrich Denker         <    [622]
60 '<                                   <    [289]
70 '<            Listing 2              <    [983]
80 '<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<    [2058]
90 '<<<<<<<<<INITALISIERUNG<<<<<<<<<<<<<<    [2502]
100 '<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<    [1883]
110 MEMORY &9FFF:LOAD"!MCHB.bin",&A000       [1228]
120 ba=&A237:mc=&A249:npr=&A25B:sjn=&A26D:  [2898]
za=&A26E:sa=&A270:basic=&A272
130 DEF FNl(x)=255 AND UNT(x):DEF FNh(x)=2  [4842]
55 AND INT(x/256)
140 '>>>>>>>>>>>>>> TITEL <<<<<<<<<<<<<<    [1629]
150 MODE 1:INK 0,0:INK 1,26:INK 2,6:INK 3,  [3635]
19:PEN 2:PAPER 0:BORDER 0
160 LOCATE 1,2:PRINT"****";:PEN 1:PRINT"'M  [4995]
C-Programme an BASIC anhaengen'";:PEN 2:PR
INT"****"
170 LOCATE 12,4:PEN 3:PRINT"BY Ulrich Denk  [2436]
er"
180 WINDOW#1,2,39,9,24:WINDOW SWAP 0,1      [2546]
190 PLOT 30,0,3:DRAWR 608,0:DRAWR 0,258:DR  [4313]
AWR -608,0:DRAWR 0,-258
200 PLOT 16,14,1:DRAWR 608,0:DRAWR 0,258:D  [3823]
RAWR -610,0:DRAWR 0,-258
210 '>>>>>>>Programmnamen eingeben<<<<<     [2347]
220 PAPER 2:PEN 1:CLS:LOCATE 4,2            [2527]
230 LOCATE 4,2:PEN 1                        [781]
240 PRINT"Bitte die Programmnamen eingeben  [4666]
"
250 PRINT CHR$(24);" (CAS: 16 Z. / DISK: 8  [2959]
 Z. + 3 Z.Ext.) ";CHR$(24):PEN 3
260 PRINT:INPUT" Basicprogramm ";s$         [1269]
270 IF s$="" THEN PEN 0:PRINT" ( DUMMY )":  [2923]
PEN 3:POKE basic,1:GOTO 290
280 adr=ba:GOSUB 570:POKE basic,0           [1386]
290 LOCATE 2,9:INPUT"MC-Programm ";s$       [2177]
300 IF s$="" THEN 290                       [599]
310 adr=mc:GOSUB 570:LOCATE 2,12            [1798]
320 LOCATE 2,12:INPUT"Neues Programm ";s$   [4262]
330 IF s$="" THEN 320                       [601]
340 adr=npr:GOSUB 570:PAPER 1:PEN 0:GOSUB   [3706]
640:IF a$<>"J" THEN 210
350 CLS:LOCATE 1,2                          [826]
360 '>>>>>>>>>MC verschieben ?<<<<<<<<<     [2177]
370 PAPER 3:CLS:LOCATE 2,2                  [1686]
380 PRINT "Soll das MC-Programm vor Aufruf  [2416]
"
390 PRINT" verschoben werden ? (J/N)"       [2299]
400 GOSUB 650:IF a$<>"J" THEN POKE sjn,0:G  [3022]
OTO 520
410 PEN 2:PRINT:PRINT" (Z)iel- u. Start-,   [3431]
nur (S)tart- oder
420 PRINT" (K)eine Adresse eingeben ?"      [3215]
430 GOSUB 650:PEN 0:PAPER 2                 [1550]
440 IF a$="K" THEN POKE sjn,3:GOTO 520      [1436]
450 IF a$="S" THEN POKE sjn,2:GOTO 480 ELS  [2956]
E POKE sjn,1
460 LOCATE 3,9:PRINT"(= Adresse an die ver  [3315]
schoben wird)"
470 LOCATE 3,8:INPUT"Zieladresse: ",z       [1857]
480 LOCATE 3,12:PRINT"( =  Calladresse im   [4870]
Basicprogramm)"
490 LOCATE 3,11:INPUT"Startadresse: ",s     [4409]
500 IF s=0 AND z<>0 THEN s=z                [1813]
510 POKE sa,FNl(s):POKE sa+1,FNh(s):POKE z  [3390]
a,FNl(z):POKE za+1,FNh(z)
520 PAPER 1:PEN 2:GOSUB 640:PEN 0           [1501]
530 IF a$<>"J" THEN 370                     [1171]
540 CLS:PAPER 0:PEN 1                       [1017]
550 IF PEEK(basic)=0 THEN CALL &A015:RUN E  [2925]
LSE CALL &A000:RUN
560 ' UNTERPRG: Programmnamen einPOKEn      [2949]
570 POKE adr,LEN(s$):s$=UPPER$(s$)          [553]
580 FOR b=1 TO LEN(s$)                      [1731]
590   POKE adr+b,ASC(MID$(s$,b,b))          [1100]
600 NEXT                                    [350]
610 POKE adr+b,0                            [209]
620 RETURN                                  [555]
630 ' UNTERPRG: Taste holen                 [2408]
640 LOCATE 3,15:PRINT"Sind alle Eingaben k  [3873]
orrekt ? (J/N)"
650 a$=UPPER$(INKEY$):IF a$="" THEN 650 EL  [2222]
SE RETURN
```

Listing VERSTECK

An unsere Leser

Die Rubrik »Leserbriefe« ist eine Einrichtung für alle Leser, die in irgendeiner Form Fragen, Probleme oder Anregungen zu Produkten, Programmierproblemen oder zu unserer Zeitschrift haben. Selbstverständlich sind wir bemüht, alle Leserfragen zu beantworten. Doch haben Sie bitte Verständnis, daß wir nicht alle eingehenden Briefe persönlich beantworten können. Oft erreichen uns mehrere Briefe zum gleichen Thema, einer davon wird dann stellvertretend für alle in unserer Zeitschrift beantwortet.

Ihre PC-Redaktion

Neue Durchwahlnummern für die Hotline

Für eilige Anfragen können Sie jetzt Ihren Redakteur direkt erreichen. Jeden Mittwoch von 17.00 – 20.00 Uhr stehen Ihnen zur Verfügung:

Claus Daschner (CPC)
☎ (0 56 51) 8 09 – 7 50

Joachim Freiburg (CPC/PC)
☎ (0 56 51) 8 09 – 7 51

Ralf Schößler (PCW)
☎ (0 56 51) 8 09 – 7 52

PS: Die Redaktion behält sich vor, Leserzuschriften in gekürzter Form wiederzugeben

In eigener Sache

Liebe Leser, liebe Leserinnen

Wir möchten Ihnen eine wichtige Angelegenheit ans Herz legen.

Jeden Tag bekommt die Redaktion eine unglaubliche Menge von Postkarten und Briefen gebracht. Darüber freuen wir uns natürlich, aber leider treten immer wieder Aspekte auf, die eine Beantwortung der Fragen unmöglich machen.

Deshalb unsere dringliche Bitte an Sie:

– Halten Sie Ihre Fragen so kurz und präzise wie möglich.

– Legen Sie bitte einen adressierten und frankierten Rückumschlag bei. Sie ersparen uns so sehr viel Arbeit, die wir dann in interessante Beiträge investieren können.

– Geben Sie immer die Redaktion und die entsprechende Abteilung an, zum Beispiel Red. PC Amstrad/Abt. CPC

Vielen Dank für Ihr Verständnis, Ihre Amstrad-Redaktion

AMX-Maus am Joyce

Vor ein paar Monaten habe ich die AMX-Maus sowie das DTP-Programm Stop Press für den Joyce PCW erstanden.

Grundsätzlich bin ich mit dieser Maus recht zufrieden. Leider erfüllt sie im Moment noch nicht ganz den Zweck, zu dem ich sie erstanden habe. Ich wollte die Maus in eigene Programme mit einbinden. Da ich die Maus nun nicht ganz so nutzen kann, wie ich mir das eigentlich vorgestellt hatte, möchte ich folgende Frage an Sie richten:

Wie kann ich die Maus in CP/M einbinden, um sie später in eigenen Programmen nutzen zu können? Sollten Sie hierzu keine Antwort wissen, würde es mir auch reichen, wenn Sie mir sagen könnten, wo im Programm Stop Press die Routinen zur Steuerung der Maus untergebracht sind.

Monika u. Bernd Martin, Morogoro

Leider wissen wir auf Ihre Frage keine Antwort. Vielleicht kann jedoch einer unserer Leser, der sich mit diesem Problem schon auseinandergesetzt hat, weiterhelfen.

Red.

Unbekanntes Format

Von Freunden bekam ich ein Diskettenlaufwerk geschenkt. Die Freude war groß, aber ... als ich das Diskettenlaufwerk näher betrachtete, stellte ich folgendes fest:

Bei dem Laufwerk handelt es sich weder um ein 3,5-Zoll- (wie angenommen), noch um ein 3-Zoll-Laufwerk. Die 3,5-Zoll-Disketten sind zu groß und die 3-Zoll-Disketten sind zu klein für das Laufwerk. Nun würde ich gerne genauer wissen, um was für ein Format es sich hierbei handelt.

Hier nun alle Angaben zu dem Laufwerk:

Firma: Tabor Corporation, Westford, MA, USA, 01886

DRIVETTE (TM) Microfloppy Disk Drive, Model TC 500.

Ein Anschreiben an die Firma führte zu keinem Erfolg.

Norbert Wimmer, Hürth

Offensichtlich ist Ihnen eines der äußerst seltenen 3,25-Zoll-Laufwerke untergekommen, welches aufgrund der stark gegen Null tendierenden Verbreitung für Sie wenig Nutzen haben wird. Behalten Sie es trotzdem, vielleicht wird der Liebhaberwert eines Tages aus Ihnen einen reichen Mann machen.

Red.

Turbo-Relocator

Der in Ausgabe 10/89 veröffentlichte Turbo-Relocator ist leider nicht vollkommen lauffähig. Der Autor hat vor dem Aufruf der Routine CAS IN OPEN &BC77 (in DATA Zeile 280 des Laders) ein PUSH IX (DD E5) und dadurch unmittelbar nach dem Aufruf ein entsprechendes POP IX (DD E1) vergessen.

Die Folge ist, daß die ermittelten Werte nicht an die entsprechenden Variablen übergeben werden, da der Zeiger IX durch die Firmware-Routine verändert wird! Abhilfe: Den Lader so korrigieren, daß das MC-Programm nicht bei Adresse &1000, sondern bei &FFC beginnt. Danach einfach die Bytewerte DD,E5 vor dem Aufruf von CAS IN OPEN (CD 77 BC – in Zeile 290) und die Werte DD, E1 gleich nach dem Aufruf (Zeile 290) einfügen. Achtung! Beim Absaven (Zeile 250) ist dadurch das Programm um vier Bytes länger – also &211.

Auch im BASIC-Programm muß die neue Adresse &FCC anstelle von &1000 in Zeile 370 eingesetzt werden. Entsprechend ist in Zeile 110 MEMORY auf &FFB zu setzen.

Aleksander Hercog, Maribor

Leserbrief von Frank Bondsa (PCI 11/89)

Die zweiten 64 kByte des CPC 6128 lassen sich zwar nicht direkt unter WordStar nutzen, es geht jedoch über einen Umweg: Wenn man diese 64 kByte unter CP/M 2.2 als RAM-Disk definiert, kann WordStar seine Overlay-Dateien auch dort suchen. Im einzelnen sieht das so aus:

In PCI 5/1987 wurde ein Programm veröffentlicht, das eine RAM-Disk von 64 kByte einrichtet. Damit können Sie die RAM-Disk als Laufwerk B: ansprechen (Voraussetzung ist, daß Sie nicht im Besitz eines zweiten Laufwerks sind). In der veröffentlichten Fassung hat sich übrigens ein Fehler eingeschlichen: Zeile 580 des zweiten BASIC-Laders muß heißen:

580 DATA 27,AE,3E,41,43, 1D,AE,AF,32,25,0357

Als nächstes müssen Sie WordStar dazu überreden, die Overlay-Dateien auf dem Laufwerk B: zu suchen. Dafür gibt es das Label DEFDSK (02dc). Dort tragen Sie für B: den Wert 2 ein (CHR$(2>), nicht die Ziffer 2). WordStar sucht nun sowohl auf dem jeweils 'eingeloggten' Laufwerk wie auch auf B:.

Zu guter Letzt müssen Sie vor dem Aufruf von WordStar die Overlay-Dateien nach B: kopieren. Das geht zum Beispiel mit folgender Submit-Datei (WS.SUB):

```
PIP B:=A:WSMSGS.OVR
PIP B:=A:WSOVLY1.OVR
WS
```

Mit 'SUBMIT WS' können Sie dann WordStar starten. Sobald das Hauptmenü erscheint, können Sie die Diskette in Laufwerk A: gegen

Ihre Datendiskette tauschen (danach neu einloggen!). Sie brauchen die WordStar-Diskette erst wieder, wenn Sie MailMerge, SpellStar oder die R-Option benutzen wollen.

Der Lohn der Mühe: WordStar baut seine Hilfsmenüs geradezu blitzartig auf. Auch die Ladezeiten für selten gebrauchte Befehle sind nicht mehr bemerkbar.

Georg Beran
Wördern

Leserbrief zu Frank Bondsa (PCI 11/89)

In der PC Amstrad 11'89 wurde von Frank Bondsa aus Lübeck die Frage gestellt, ob man nicht die WordStar-Overlay-Dateien in der zweiten Speicherbank des CPC 6128 unterbringen könne. Das ist möglich, allerdings nur unter CP/M 2.2, das ja auf einer der beiden Systemdisketten dem Computer beiliegt. Man braucht nur noch ein Programm, das die zweite Speicherbank als Ramdisk einrichtet – ich besitze ein solches Programm. Dieses Listing erstellt eine COM-Datei, mit der das CP/M 2.2 verändert wird, so daß es die zweite Speicherbank als Laufwerk B: anspricht. Mein WordStar, das für CP/M plus vorgesehen ist, wurde umgepatcht, so daß es mit den geänderten Bildschirmaufbau unter CP/M 2.2 läuft (Beschreibung im Sonderheft 2/86, Seite 91).

Unter diesen Voraussetzungen muß WordStar nur noch uminstalliert werden: Install aufrufen, "N" (sofern es keine Erstinstallation ist) eingeben, Punkt B anwählen und bei allen folgenden Menüs "U" für "No change" (keine Veränderung) eingeben, und bei "O.K." mit "Y" antworten. Schließlich wird gefragt: "Are the modifications now complete?" – dann ein "N" eingeben für "Additional patches". Nun wird der Eintrag geändert, der WordStar vermittelt, wo es die Overlay-Dateien suchen soll. Geben Sie ein: DEFDSK: , für Laufwerk B: 02.

Um den Aufbau der Menüs noch zu beschleunigen, mit "DEL3:" einen Wert zwischen "02" und "05" eingeben.

Künftig braucht man eine Diskette mit den beiden Overlay-Dateien, die man vor dem Arbeiten mit WordStar in die RAM-Disk kopiert, und eine Diskette mit "WS6128.COM", auf der auch die Texte gespeichert werden. Ein Nachteil ist allerdings, daß WordStar unter CP/M 2.2 nur etwas mehr als einzeilige Blöcke verschieben kann. Die Umlaute müssen mit einem speziellen Programm vom Betriebssystem aus eingestellt werden, und die Tastaturbelegung mit Setup ist wesentlich weniger komfortabel als unter CP/M plus. Für eingehendere Fragen zu der RAM-Disk und dem Umpatchen von WordStar stehe ich den interessierten Lesern telefonisch zur Verfügung.

Tel.: 04534/1347
Manfred Wilhelm,
Lütjensee

Diese Briefe haben wir stellvertretend für viele Antworten veröffentlicht. Da kann man unserer Leserschaft nur ein großes Lob aussprechen, die uns und unseren Lesern weitergeholfen hat.

Red.

Quadratische Gleichungen

Seit Dezember 1985 bin ich Leser Ihrer Zeitschrift, und ich finde sie ausgezeichnet. Vor allem die 1-kByte-Programme machen mir ziemlich Spaß. Nur als ich jetzt ein Programm aus Heft 7/89 (Seite 56) abgetippt habe, fiel mir ein zusätzlicher Lösungsweg ein. Der Zusatz ermöglicht komplexe Lösungen. Es werden hierbei zwei Zeilen geändert und drei hinzugefügt.

```
Zeile 210 IF d<0 THEN PRINT
"Die Lösungsmenge ist konjugiert Komplex !":GOSUB 280
Zeile 230 IF d<0 THEN PRINT
x11;"+";x12;"i ;"; x11;"-";
x12;"i}"
Zeile 280 d=ABS (d)
Zeile 290 x11=ROUND ((1/(2*a))
*SQR (d),2)
Zeile 300 RETURN
```

Egon Soukal,
Gefrees

Mehr Text mit ConText auf dem CPC 6128

Nach wie vor ist für mich der CPC 6128 das ideale Gerät und ConText das bequemste Textsystem. Leider nutzt es nicht die starke Seite des 6128, nämlich die RAM-Bank 0 von 4 Blöcken zu je 16 kByte. Doch nach einem kurzen Versuch wurde mir plötzlich klar, wie man statt der üblichen fünf Seiten von je 80 x 64 Bytes, zusammen also 25600 Bytes, die große Menge von 90112 Bytes zur Verfügung haben kann (wenn auch mit gewissen Einschränkungen).

Bekanntlich kann man durch bestimmte OUT-Befehle Teile der Bank 0 an die Stelle der sonst für den Computer allein "sichtbaren" Blöcke der Bank 1 einblenden. Am besten wird jeweils einer der 4 Blöcke (Nr. 0 bis 3, jeweils 16 kByte) von Block 0 anstelle des Blocks 1 von Bank 1 eingeschaltet. Es handelt sich hierbei um die RAM-Adressen &4000 bis &7FFF. Die Adressen bleiben für den CPC immer dieselben, gleich welcher Block in diesem Bereich eingeblendet wird.

Wie berühren diese Adressen nun die einzelnen Teile von ConText – BASIC-Programm, Maschinencode und Freiraum für Text? Wir haben Glück! Sie liegen mitten im Textbereich, von &2F00 bis &92FF.

Zum Umschalten eines der Blöcke wählt man mit dem Pfeil im linken Viereck ganz unten 'BASIC' an. Das Programm schaltet ab, und es erscheint das READY-Signal. Nun gibt man im Direktmodus einen der nachstehenden Befehle ein:

```
OUT &7F00,&C4
(um Block 0 von Bank 0 einzublenden)
OUT &7F00,&C5
(um Block 1 von Bank 0 einzublenden)
OUT &7F00,&C6
(um Block 2 von Bank 0 einzublenden)
OUT &7FFF,&C7
(um Block 3 von Bank 0 einzublenden)
OUT &7FFF,&C0
(um Block 1 von Bank 1 einzublenden und damit zum ursprünglichen Zustand zurückzukehren)
```

Fügt man an den eingetippten Befehl noch einen Doppelpunkt und RUN an, erscheint nach ENTER das ConText-Menü umgehend wieder, und auch der vorherige Text ist noch da, weil er sich ja oberhalb von HIMEM befindet.

Hiernach ist der betreffende Block im Raum von Seite 1, Zeile 58, Spalte 49 über die kompletten Seiten 2, 3 und 4 bis zur Seite 5, Zeile 4, Spalte 16 sichtbar; allerdings nicht ganz sauber, doch lassen sich die anfänglich sichtbaren Streifen löschen.

Keine Angst, der ursprünglich in diesem Bereich vorhandene Text ist auch hier nicht verloren, er erscheint auf den letzten der angegebenen OUT-Befehle wieder. In die neuen Bereiche können Sie nun zum Beispiel Texte aus den Festbereichen von Seite 1 oder 5 kopieren; um etwa Textauszüge zu erstellen, oder Sie können natürlich auch einen neuen Text hineinschreiben. Der in die vier Blocks von Bank 0 eingegebene Text überdauert sogar einen Reset durch den "Dreifingergriff".

Und die eingangs erwähnten Einschränkungen?

1. Bei seitenorientierter Textgestaltung sind außer im Haupttext die Streifen auf Seite 1, Zeilen 58 bis 64 und auf Seite 5, Zeilen 1 bis 4 nur für Notizen und dergleichen nutzbar, zum Beispiel zur Kennzeichnung des eingeblendeten Blocks. Immerhin handelt es sich bei diesen Streifen für alle vier Zusatzblocks zusammen nur um etwa 3 kByte im Vergleich zum Gesamtraum von 90112 Bytes.

2. Der Computer kann nur den Text als Datei abspeichern, den er "auf einen Blick" erfassen kann. Der Text in einem gerade nicht sichtbaren Block kann erst dann abgespeichert werden, wenn er eingeblendet worden ist. Das heißt, die Datenlänge kann durch die Nutzung von Bank 0 nicht von fünf auf neun Seiten vergrößert werden! Vielleicht gibt sich einmal jemand daran, die OUT-Befehle in das ConText-Programm zu intergrieren. Ich komme jedenfalls auch so ganz gut zurecht, und ich wünsche Ihnen gleichfalls ein gutes Arbeiten mit dem erweiterten ConText.

Helmut Felber, Köln

Es ist schon manchmal beachtlich, welch naheliegende Lösungen auf komplizierte Fragestellungen lange auf ihre Entdeckung warten. In jeden Fall danken wir Herrn Felber, für diesen hilfreichen Tip.

Red.

Problem mit Turbo Pascal

Seit geraumer Zeit beschäftige ich mich mit Turbo Pascal 3.0 und Turbo Pascal 5.0.

Mein Interesse gilt hauptsächlich der mathematischen Programmierung. Dabei ergibt sich häufig das Problem der Bearbeitung von Funktionen. Wie Sie wissen, stellt das Einbinden einer Funktion so lange kein Problem dar, bis man das Programm in compilierter Form nutzen will. Dann nämlich ist es nicht mehr möglich, einen Funktionsterm direkt einzugeben.

Doch gerade dies wäre wünschenswert, um Programme auch Leuten, die kein Turbo Pascal besitzen, zugänglich zu machen.

Man benötigt also eine Prozedur, die einen als String eingegebenen Funktionsterm untersucht bzw. bearbeitet; so daß man das "Ergebnis" ähnlich einer Funktion aufrufen kann. Leider bin ich kein ausgesprochener Pascal-Freak, so daß diese Hürde für mich bis jetzt unüberwindbar war.

Vielleicht kann ein Pascal-Spezialist einem Nicht-Spezialisten einen Trick verraten.

Mario Bugiel, Fischingen

Haben Sie etwas Geduld! Als Antwort innerhalb der Leserbriefe erscheint uns eine nachvollziehbare Lösung zu umfangreich. Wir werden uns des Problems annehmen und in einer der nächsten Ausgaben eine ausführliche Lösung anbieten.

Red.

Betrifft: Zeitschriftenverwaltung aus Heft 11/89

In Ihrer Zeitschrift PC Amstrad International wurde mein Programm "Zeitschriftenverwaltung" veröffentlicht. Nachträglich habe ich leider einen kleinen Fehler entdeckt.

In der Bildschirmmaske wird der Strich, der die beiden Informationsfelder 'Freie Datensätze' und 'Eingelesene Datei' trennt, teilweise entfernt, wenn man sich im Menüpunkt Nr. 5 befindet, einen Datensatz bearbeitet, den Schlüssel '1' setzt und zum nächsten Datensatz blättert. Dies läßt sich verhindern, indem man in der Zeile 1470 eine kleine Änderung vornimmt. Und zwar:

.... PLOT 199,366: DRAW 199,382 ... in ... PLOT 201,366:DRAW 201,382 ...

Bernd Hartenberger, Ansbach

Scanner für den CPC?

Gibt es einen Hardwarezusatz (Scanner), mit dem man einzelne Textzeilen und Barcodes lesen und diese Daten dann in einem eigenen BASIC- oder dBaseprogramm weiterverwenden kann?

Wenn ja, zu welchem Preis und wo kann ich dieses Gerät beziehen?

Markus Drexelius, Usingen

Zur Zeit gibt es nur einen einzigen Scanner, der Grafik in den CPC übertragen kann. Er heißt 'Dart-Scanner' und wird auf dem Druckerkopf eines DMP 2xxx gespannt. Nach Einlegen einer Grafikvorlage in den Drucker wird dieser mit dem Scanner abgetastet und punkteweise an den CPC gegeben. Allerdings ist dieses System nicht für eine Texterkennung geeignet. Bei Listings, die in unserer Zeitschrift abgedruckt werden, empfehlen wir daher den Erwerb der Databox. Informationen zum Scanner bekommen Sie im Fachhandel oder bei

Weeske Computer-Elektronik
Potsdamer Ring 10
7150 Backnang
Tel.: 07191/1528-29

Red.

Assemblerecke (PCI 10/89)

In der Tabelle 2 auf Seite 41 hat sich ein kleiner Fehler eingeschlichen. In der Tabelle steht, daß bei RAM-Konfiguration 3 der Speicherbereich ab #4000 nicht belegt sei. Dies ist falsch; in dem Bereich liegt in diesem Falle Bank 3.

Sönke Ostertun, Henstedt

DIN A4 für Artworx

Bei der Änderung von Heiko Glöde, welche in Augabe 9/89 auf Seite 58 abgedruckt ist, ergeben sich folgende 'Mängel':

1. Die Angaben (in cm) gemessen am Bildanfang DIN A5 stimmen mit der Vorgabe nicht überein. So wurde zum Beispiel der Ausdruck bei vorgewählten 10 cm schon ab zirka 7,5 cm in DIN-A4-Format gedruckt.

Bei 3 cm beginnt ab zirka 6 mm der Großformatabdruck.

2. Eine Eingabe von 1 cm führte zur Fehlermeldung 'Improper Argument in Zeile 224'. Es sollte eine negative Zahl (-6) gepoket werden.

Als Lösung ist die Änderung der Zeile 223 anzubieten:

```
223
a=INT(a*5.67+1):POKE
&1601,160-a
```

Siegfried Meier, Eutin

Erfahrungsaustausch BTX

Wer kennt sich mit BTX und dem Betreiben des Dataphones S21-S23d aus, was laut unzureichendem Betriebsbuch BTX-fähig sein soll?

Für einen Erfahrungsaustausch wäre ich sehr dankbar!

*Martin Frank
Neuestr. 23
2200 Elmshorn*

FAst BAsic COMpiler



BASIC-Compiler für CPC 464/664/6128

Der Turbo-Antrieb für Ihre BASIC-Programme!

Haben auch Sie sich schon immer gewünscht, daß Ihre selbstgeschriebenen BASIC-Programme schneller laufen? Mit dem BASIC-Compiler von DMV ist das nun kein Problem mehr, denn

- **der Compiler hat den vollen Sprachumfang des BASIC 1.1 (CPC664/6128).**
- **das compilierte Programm ist auf jedem CPC lauffähig.**
- **unterstützt Integer- und Fließkomma-Arithmetik.**
- **ist kompatibel zu Vortex-Peripherie inkl. Nutzen der RAM-Disk.**
- **Programme, die spezielle BASIC-1.1.-Befehle beinhalten, sind auch auf dem CPC464 lauffähig (außer FILL und MASK).**
- **der Compiler arbeitet unter CP/M, das heißt, alle CP/M-Dienstprogramme können genutzt werden.**
- **bis 17 kByte Quellcode können problemlos compiliert werden.**
- **einzelne Programmteile können ebenfalls compiliert werden (z.B. wichtig bei Nachladeprogrammen).**
- **die ausführliche deutsche Bedienungsanleitung macht Sie auf einfache Weise mit dem Umgang des Compilers vertraut.**
- **viele Beispielprogramme veranschaulichen die Arbeitsweise des Compilers und zeigen die Geschwindigkeitsvorteile auf.**
- **das Programm ist in 100% Maschinencode geschrieben.**

Der BASIC-Compiler ist nur auf 3"-Diskette erhältlich.

Best.-Nr.: 209 **Preis 49,– DM***

Bitte benutzen Sie die Bestellkarte.

Möchten Sie gerne in Assembler programmieren ?

CPC Assembler Software und Trainingsbuch

Das Buch:

führt den Anfänger schrittweise in die Programmierung des Z80 ein. Dabei werden Vorkenntnisse nicht vorausgesetzt. Nach der Lektüre des Buches sind Sie mit dem Befehlssatz des Prozessors wie auch mit den Adressierungsarten vertraut. Anhand einer Fülle von Programmbeispielen, die speziell auf den CPC 464 zugeschnitten sind, lernen Sie, nützliche Routinen in Maschinensprache zu entwickeln, die auf Ihrem Rechner sofort lauffähig sind. Dazu erfahren Sie, wie Sie die im ROM des Schneider CPC vorhandenen Hilfsroutinen sinnvoll für die Lösung eigener Probleme einsetzen können. Darüber hinaus lernen Sie einige spezielle Programme zur Erweiterung des Betriebssystems mit leistungsfähigen grafischen Befehlen kennen, die Sie in BASIC verwenden können, z.B. TRIANGEL, BOX und CIRCLE. Ein umfangreicher Anhang mit Erläuterungen des Befehlssatzes, des Assemblers sowie einer Vielzahl kommentierter Einsprungadressen des Betriebssystem-ROMs rundet das Buch ab.

Die Software:

besteht aus einem menügesteuerten Z80 - Assembler.
Mit seiner Hilfe können Sie nicht nur die im Buch erläuterten Beispielprogramme editieren und in Maschinensprache übersetzen, sondern auch selbst entworfene Programme entwickeln und testen. Für Ihre Arbeit stehen Ihnen Funktionen zur Verfügung wie **Einfügen – Löschen – Ändern von Programmzeilen – Abspeichern und Laden von Programmen auf Datenträgern wie Kassette und Diskette – Ausgabe von Listings auf Bildschirm und Drucker – Zahlenkonvertierung von der Menü-Ebene aus – Durchführung von Zwischenrechnungen mit binären, hexadezimalen und dezimalen Zahlen.** Weiterhin erhalten Sie ein Trainingsprogramm, mit dessen Hilfe Sie den Umgang mit verschiedenen Zahlenformaten vertiefen können, und Programme zur Demonstration der grafischen Erweiterungen des Betriebssystems.

Buch mit Kassette Best.-Nr.: 446 **Preis 39,– DM***

Buch mit Diskette Best.-Nr.: 447 **Preis 49,– DM***

*Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege





JUCA

Ein Adressverwaltungs-Programm für dBase II-Fans (2)

Nachdem im ersten Teil der Serie grundsätzliches zur Entstehung und zum Aufbau von JUCA nachzulesen war, werden im zweiten Teil die JUCA-Module MV1000 (Verwalten) und MV2000 (Anzeigen) beschrieben. Die Dateien auf der DATABOX sorgen dafür, daß Sie JUCA bereits mit Daten füttern und diese verwalten und anzeigen können.

Wenn Sie die Start-DISK wie im Teil 1 unter Ziffer 01. und 02. beschrieben erstellt haben, kopieren Sie von der DATABOX zu diesem Heft die JUCA-Programmdateien mit der Endung .CMD auf die Seite 2 von DISK A und die JUCA-Datendateien mit der Endung .DBF und .NDX auf die Seite 1, Gruppe 0, der DISK B. Sollten Sie aufgrund der letzten Folge auf DISK A oder B bereits Datendateien generiert haben, so löschen Sie diese, bevor Sie die jetzt zur Verfügung gestellten Dateien (.DBF und .NDX) dort ablegen. Die in Abbildung 3 (Teil 1) genannten Dateien MV3000.CMD, MV3100.CMD, MV3200.CMD und MV3300.CMD stehen Ihnen erst nächsten Monat zur Verfügung; Sie dürfen daher die Optionen, die etwas mit Drucken zu tun haben, noch nicht benutzen.

So starten Sie JUCA

03. Geben Sie Seite 1 der DISK A ins LW A> und booten Sie CP/M PLUS durch ein Reset (gleichzeitiges Drücken von [SHIFT] [EXTRA] [EXIT]). Einige Dateien werden nun ins LW M> kopiert und Sie werden aufgefordert, DISK A im LW A> umzudrehen. Nach [RET] sorgt die Anweisung in der Datei PROFILE.SUB dafür, daß auch die JUCA Programmdateien ins LW M> übertragen werden. Danach wird die Datei MVA.SUB im LW M> gestartet; sie sorgt für die notwendige Umstellung von Tastatur und Drucker (Hinweis: Sehen Sie sich die Dateien DB.KEY, STANDARD.KEY, MV.PTR und STANDARD.PTR mittels TYPE (zum Beispiel TYPE MV.PTR [RET]) einmal genauer an; Erläuterungen zu den verwendeten Steuer-Codes sind im JOYCE-Benutzer-Handbuch (SCHNEIDER) zu finden. Danach erfolgt der Start von dBase II und JUCA.

04. Im Startmenü werden Sie zum Einlegen der Daten-DISK (DISK B) und zur Eingabe der LW-Bezeichnungen aufgefordert. Haben Sie einen PCW8256 mit einem LW und erweiterter RAM, so nehmen Sie DISK A aus dem LW und geben die für LW A> formatierte DISK B, Seite 1, hinein. Die Voreinstellungen ('A' und 'A') werden für das Programm-LW geändert (es wird [M] eingegeben) und bleiben für das Daten-LW unverändert (es wird [A] eingeben). Wenn Sie einen PCW8512 mit 2 LW benutzen, lassen Sie die DISK A im LW A> und geben DISK B ins LW B>. Als Programm-LW werden [M] und als Daten-LW [B] eingegeben. Nach der zweiten Eingabe fährt das Programm automatisch fort. (Hinweis: Beim Programmstart wird auf DISK A, Seite 1, eine Datei SYSIN59.$$$ angelegt, die dort getrost belassen werden kann; sie hat für JUCA keine Bedeutung und stört nicht.)

05. Das Programm prüft nun, ob sich auf DISK B Datendateien befinden. Werden keine Datenbanken vorgefunden, so erzeugt das Programm diese, wenn Sie nach dem Hinweis 'Keine Datendateien auf dem Laufwerk! Dateien anlegen J/N?' Ihr Ja-Wort geben, also 'J' drücken; das tun Sie nun aber nicht! Sie landen nun im JUCA-Hauptmenü (Abbildung 2, Teil 1). Da wir zur Erleichterung zunächst nur auf Musterdatenbanken zurückgreifen, kommen wir später noch einmal hierauf zurück (Schritt 25).

06. Es bedarf an dieser Stelle des Hinweises, daß die Stapeldatei MVA.SUB, mit der JUCA von LW M> aus gestartet wurde, aktiviert bleibt. Sie wird weiter abgearbeitet, wenn Sie im JUCA-Hauptmenü [0] drücken, die Frage nach der Durchführung von PACK mit [N] quittieren (siehe Schritt 24), JUCA also verlassen und nach CP/M PLUS zurückkehren. Nachdem Sie diese Eingaben getätigt haben, erfolgt die Rücksetzung von Tastatur und Drucker zur Grundeinstellung. Sie sollten das jetzt ausprobieren.

07. Der erneute Start vom LW M> aus geschieht mit SUBMIT MVB [RET] beziehungsweise MVB.SUB [RET]. Sehen Sie sich die unterschiedliche Anweisung in beiden SUB-Dateien mit TYPE an!

Die Datei MVC.SUB wird nur benutzt, um Änderungen im den Programmdateien vorzunehmen. Falls Sie das möchten, so geben Sie im LW M> SUBMIT MVC [RET] beziehungsweise MVC.SUB [RET] ein. Das sollten Sie jedoch nur auf einer Sicherheitsko-

pie von JUCA tun und nur dann, wenn Sie dBase II schon ausreichend beherrschen beziehungsweise erlernen wollen!

Das Teilprogramm 'Verwalten'

08. Wählen Sie im Hauptmenü die Option [1] (Verwalten), im folgenden Menü [1] (Verwalten Personenstamm) und im nächsten [1] (Personen anlegen). Auf dem Bildschirm werden die für die Eingabe zur Verfügung stehenden Felder eines Datensatzes invers dargestellt (Abbildung 4). Diese entsprechen der bestehenden Datensatzstruktur, wie sie weiter oben aufgezeigt wurde. Geben Sie nun einige Datensätze ein (nach jeder Zeile [RET]), zum Beispiel so:

```
Sitzfleisch
Helmut
Politiker
Klöngelpfad 99 b
7341 Stuckard 1
077-1299560
0771-168-234
JOY1 RAMER
LOCO1 LOTTO JPRO1
BASIC
```

das nächste [RET] bietet Ihnen einen neuen Datensatz zur Eingabe an:

```
Qualm
Walter
Zigarrendreher
Dunststieg 21
4077 Bad Partitur 2
034-617137
034-617138
JOY2
LOCO2 DRDRA DRGRA LOFI2
CBASC DBASE
```

und so weiter.

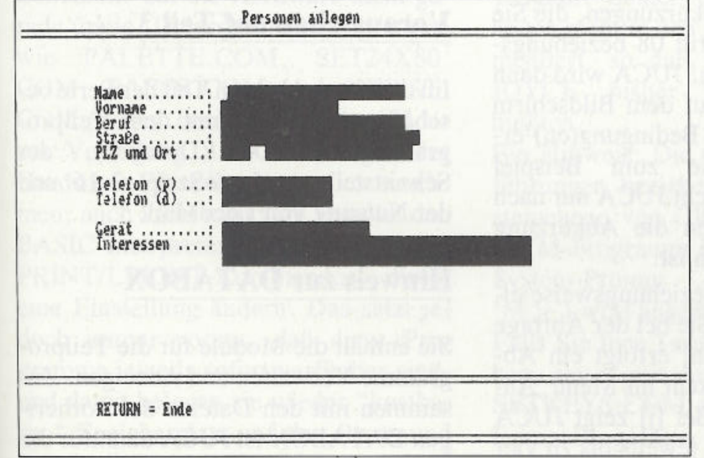
Benutzen Sie bei der Eingabe für die Felder 'Gerät', 'Interesse 1' und 'Interesse 2' Abkürzungen von maximal fünf Zeichen, zum Beispiel 'JOY1' für 'JOYCE PCW8256 mit 1 LW', 'LOCO2' für 'LocoScript 2' oder 'DRGRA' für 'DR GRAPH'. Zwischen zwei Abkürzungen geben Sie jeweils ein Leerzeichen ein. Achten Sie darauf, daß Sie bei einigen Datensätzen die gleichen Abkürzungen eingegeben haben; weiter unten finden Sie die Erklärung dafür. Nach Eingabe der Datensätze und [RET] in Datensatzfeld 'Interesse 2' wird erneut ein Datensatzfeld zur Eingabe eines weiteren Datensatzes angeboten. Ohne eine Eingabe zu machen, drücken Sie nun solange [RET], bis nach dem letzten Feld das Abspeichern und die Rückkehr ins Menü 'Verwalten Personenstamm' erfolgen. Anstelle von [RET] können Sie auch die dBase-Befehle [ALT] [W] für 'Abspeichern' beziehungsweise [ALT] [Q] für 'Abbrechen ohne Abspeichern' benutzen.

09. Als nächstes wählen Sie die Option [2] (Personen anzeigen, bearbeiten, löschen). Nun erscheint als erste Adresse Hochschulprofessor Axelschweiß (Abbildung 5). Sie suchen aber den Datensatz 'Sitzfleisch'. Da der Politiker Sitzfleisch nicht in Ihr politisches Weltbild paßt, soll er aus der Kartei gelöscht werden. Sie wählen [S] (Suchen) und geben 'Sitzfleisch' ein. Nach [RET] wird der Datensatz angezeigt. Nach [L] (Löschen) erhalten Sie den Hinweis 'Datensatz wird gelöscht'. Danach drücken Sie [RET].

10. Suchen Sie jetzt über [S] den Datensatz 'Qualm' und gehen Sie über [B] (Bearbeiten) in den Editiermodus. Setzen Sie den Cursor hinter 'JOY2' und ergänzen die Eintragung so, daß dort 'JOY2 L5.25' ('JOY2' für 'PCW 8512', 'L5.25' für 'Externes 5 1/4-Zoll-LW') stehen. Die Leerzeichen zwischen den Abkürzungen sind wichtig, damit das Programm sie später wiederfindet! Drücken Sie dann solange [RET], bis die inverse Anzeige verschwindet beziehungsweise benutzen Sie wieder die dBase-Befehle [ALT] [W] oder [ALT] [Q]. Mit [0] [0] kehren Sie ins Menü 'Verwalten' zurück.

11. Wählen Sie [2] (Verwalten Gruppen) und danach [1] (Gruppen anlegen). Geben Sie als Gruppen-Nr. '3' und als Kennzeichnung der Gruppe 'LocoScript-Experten' ein. Die Frage nach der richtigen Eingabe beantworten Sie mit [J] und kehren ins Menü 'Verwalten Gruppen' zurück. Dieses Arbeiten mit Gruppen ist eine Besonderheit von JUCA: Sie brauchen bei der Eingabe von Adressen keine Rücksicht darauf zu nehmen, ob es nun Privat-, Firmen- oder sonstige Adressen sind und in welcher Reihenfolge Sie sie eintippen, denn über die Gruppen können Sie die Adressen später nach Belieben ordnen, Sie sollten Gruppen jedoch erst dann anlegen, wenn Sie sie wirklich benötigen, damit nicht unnötiger Speicherplatz belegt wird.

12. Über Option [3] erhalten Sie Gelegenheit, bestimmte Personen einer Gruppe zuzuordnen. Als Gruppe geben Sie '3' ein. Blättern Sie dann die Datensätze mit [+*] beziehungsweise [−] durch. Erscheint der Datensatz 'Tastenschlag', so drücken Sie [Z] (Zuordnen) und nach dem Hinweis 'Person wurde der Gruppe zugeordnet' [RET]. Den Namen 'Tastenschlag' können Sie auch über [S] auch suchen lassen. Verfahren Sie mit zwei bis drei Datensätzen, bei denen im Feld 'Interessen' die Abkürzung 'LOCO2' zu sehen ist, entsprechend. Mit [0] kehren Sie zum Menü 'Verwalten Gruppen' zurück.



```
Personen anlegen

Name .........:
Vorname .......:
Beruf .........:
Straße ........:
PLZ und Ort ...:

Telefon (p) ...:
Telefon (d) ...:

Gerät .........:
Interessen ....:

RETURN = Ende
```

Abbildung 4: Eingabemaske im JUCA-Modul MV1110 (Verwalten / Personen anlegen)

```
Personen anzeigen, bearbeiten, löschen

Name .........: Axelschweiß
Vorname .......: Adolfo
Beruf .........: Hochschulprofessor
Straße ........: Prüfungsweg 77
PLZ und Ort ...: 3001 Hahnover 3

Telefon (p) ...: 0918-92839192
Telefon (d) ...: 0918-1111-234

Gerät .........: JOY1 RAMER L3.5
Interessen ....: LOCO2 DBASE WORDS SPIEL ASTRO

+ = Vorblättern    - = Zurückblättern    S = Suchen nach Nachnamen
B = Bearbeiten     L = Löschen           0 = Ende
```

Abbildung 5: Bildschirmdarstellung im JUCA-Modul MV1120 (Verwalten / Personen anzeigen, bearbeiten, löschen)

```
                      Anzeigen Personen
Name                   Vorname          Beruf                  Tel. priv.
PLZ  Ort               Strasse                 Gerät           Tel. dienst.
Interesse 1                             Interesse 2

Axelschweiß            Adolfo           Hochschulprofessor     0918-92839192
3001 Hahnover 3        Prüfungsweg 77          JOY1 RAMER L3.5 0918-1111-234
LOCO2 DBASE WORDS SPIEL ASTRO

Geißbart               Wilhelm-Ottokar  Förster                063120-13
9000 Bärlin 73         Am Wald 100             JOY2            063120-26
LOCO2 DBASE DRGRA                       TURPA

Wunderlich             Clodhilde        Hausmaklerin           0817-718293
5722 Freifurt/Mein     Gebührenplatz 6         JOY2            0817-718239
LOCO2 INOFO DBASE                       SMALC

     Gruppe Nr.:                                         0 = Ende :
     Name  -von: Axelschweiß        -bis: Wunderlich
     Interesse : DBASE                                 okay (j/n) :

                 Keine weiteren Personen anzuzeigen !
```

Abbildung 6: Bildschirmdarstellung im JUCA-Modul MV2100 (Anzeigen / Anzeigen Personen)

```
                   Anzeigen Abkürzungen
        Abkürzung     Erläuterung
          JPRO3       JOYCE Programmsammlung III (DMV)
          JPRO4       JOYCE Programmsammlung IV (DMV)
          JUCA        JOYCE-USER-CLUB-ADRESSVERWALTUNG (ERTEL)
          KFZAB       Kfz-/Verbrauchsabrechnung (KEMPENICH)
          KICK        CP/M PLUS Benutzeroberfläche
          KNIFE       Knife/Knife Plus engl. Disketten-Editor
          L3.5        3 1/2"-Laufwerk (extern)
          LASER       Laserdrucker
          LOCFI       LocoFile Dateiverwaltung für LOCO2
          LOCFO       LocoFont Zusatzschriften für LocoScript2

                                                      0 = Ende :
        Abkürzung -von:        -bis:
                                                   okay (j/n) :
```

Abbildung 7: Bildschirmdarstellung im JUCA-Modul MV2300 (Anzeigen / Anzeigen Abkürzungen)

13. Drücken Sie [4] (Personen aus einer Gruppe löschen) und geben Sie als Gruppe '3' ein. Die dieser Gruppe zugeordneten Datensätze werden daraufhin angezeigt. Sind mehr Datensätze vorhanden, als im Anzeigebereich dargestellt werden können, so werden weitere Datensätze erst nach [RET] angezeigt. Das Löschen eines Datensatzes geschieht, indem [L] gedrückt und die vor dem Datensatz angezeigte Nummer des Datensatzes eingegeben wird. Versuchen Sie das mit einer Eintragung und kehren Sie nach dann mit [RET] und [0] in die Gruppenverwaltung zurück.

14. Probieren Sie nun noch die Option [2] aus und löschen Sie die Gruppe '3'. Ich verzichte hier auf die weitere Beschreibung, da sich das analog zu dem bisher Gesagten abspielt. Nach dem Löschen der Gruppe kehren Sie mit [0] [0] ins Menü 'Verwalten' zurück.

15. Wiederholen Sie die Schritte 11 bis 13; lassen Sie die Gruppe 3 dann aber bestehen. Kehren Sie über [0] zum Menü 'Verwalten' zurück.

16. Befassen Sie sich nun mit der Option [3] (Verwalten Legende). Geben Sie [f3] ein und wählen Sie im nächsten Menü die Option [1] (Abkürzungen anlegen); die inverse Datensatzstruktur wird daraufhin zur Eingabe angeboten. Im Feld 'Abkürzung' geben Sie zum Beispiel 'LOCO2' ein, im Feld 'Erläuterung' dann 'PCW8512 mit 2 LW'. Nach [RET] verfahren Sie entsprechend mit den nächsten Abkürzungen und deren Erläuterungen. Nachdem Sie mit zweimal [RET] einen Datensatz ohne Eingabe durchlaufen haben, kehrt das Programm zum Menü zurück.

17. Über die Option [2] ('Abkürzungen anzeigen, bearbeiten, löschen') können Sie die zuvor getätigten Eingaben ansehen, suchen, gegebenenfalls auch korrigieren, falls Sie Fehler produziert oder Abkürzungen doppelt eingegeben haben. Verfahren Sie dabei wie unter Schritt 09 beschrieben. Kehren Sie anschließend mit [0] [0] [0] ins Hauptmenü zurück.

So arbeiten Sie mit dem JUCA-Teilprogramm 'Anzeigen' (MV2000)

18. Vom Hauptmenü aus wählen Sie [2] und danach [1], um in das Menü 'Anzeigen Personen' zu gelangen. Falls Sie noch keine Gruppe angelegt haben, drücken Sie einfach [RET], anderenfalls geben Sie die gewünschte Nummer der Gruppe ein, zum Beispiel [3]. Wollen Sie nur einen Ausschnitt der Adressenkartei ansehen, so geben Sie unter den entsprechenden Feldern die gewünschten Namen ein, zum Beispiel von 'Mutzenbecher' bis 'Tastenschlag'; ansonsten gehen Sie mit [RET] [RET] zum Feld 'Interesse'. Hier können Sie eine der Abkürzungen, die Sie benutzt haben (Schritt 08 beziehungsweise 16), eintippen. JUCA wird dann nur die Adressen auf dem Bildschirm anzeigen, die diese Bedingung(en) erfüllen: Geben Sie zum Beispiel 'LOCO2' ein, so sucht JUCA nur nach Adressen, bei denen die Abkürzung 'LOCO2' vorhanden ist.

Nach der Eingabe beziehungsweise einem [RET] landen Sie bei der Abfrage 'okay (j/n):'. Bei [n] erfolgt ein Abbruch und die Rückkehr ins Menü 'Anzeigen Personen'. Bei [j] zeigt JUCA auf dem Bildschirm jeweils bis zu vier Datensätze an (Abbildung 6). Weitere erscheinen gegebenenfalls nach [RET]. Ist der letzte Datensatz angezeigt, so wird der Hinweis 'Keine weiteren Personen anzuzeigen!' eingeblendet. Mit [RET] kehren Sie ins Menü 'Anzeigen' zurück. (Hinweis: Sehen Sie sich das Verknüpfen von Bedingungen bei der dBase-Programmierung in der Datei MV2100 an!)

19. Über die Option [2] (Anzeigen Gruppen) erhalten Sie Hinweise zu den bereits vorhandenen Gruppen und der Anzahl der darin enthaltenen Datensätze. Mit [RET] gelangen Sie zurück zum Menü 'Anzeigen'.

20. Option [3] (Anzeigen Abkürzungen) gestattet es Ihnen, die gespeicherten Abkürzungen anzusehen. Nach [RET] [RET] [j] werden alle vorhandenen Abkürzungen und Erläuterungen gezeigt; mit [RET] geht's weiter. Wenn Sie in den Feldern 'Namen von:' und 'Namen bis:' Eingaben tätigen, zeigt JUCA nur die gewünschten Abkürzungen an (Abbildung 7). Mit [0] [0] kommen Sie wieder zum Hauptmenü zurück.

Vorausschau auf Teil 3

Im dritten und letzten Teil der Serie beschäftigen wir uns mit dem Teilprogramm MV3000 (Drucken), der Schnittstelle zu LocoScript 2.16 und der Nutzung von LocoMail.

Hinweis zur DATABOX

Sie enthält die Module für die Teilprogramme Verwalten und Anzeigen. Zusammen mit den Dateien der vorherigen DATABOX ist JUCA dann für die Datenaufnahme bereit.

(Wolfgang Ertel/Detlef Gehring/rs)



Tastenspiele

Bildschirm und Druckerfunktionen unter CP/M

Wer unter CP/M Plus die Möglichkeiten des JOYCE-Systems (PCW 8256/8512) voll ausschöpfen will, steht immer wieder vor dem Problem, die Grundeinstellung von Drucker beziehungsweise Monitor an die eigenen Wünsche anzupassen. Gerade wenn man "mal eben" ein Listing ausdrucken möchte oder sich "schnell mal" die Datei XYZ auf dem Bildschirm ansehen will, ist es wünschenswert, möglichst ohne große Umstände (sprich: Programme) die bestehende Einstellung des benutzten Ausgabegerätes zu ändern.

Da man sich gewöhnlich, wenn die oben angenommenen Einstellungen vorgenommen werden sollen, auf der untersten Ebene von CP/M, also unter der Aufsicht des CONSOLE COMMAND PROCESSORS (CCP) befindet, ist die Auswahl der möglichen Einflußnahme auf die Hardware nicht gerade üppig. Diverse Dienstprogramme wie PALETTE.COM, SET24X80.COM, PAPER.COM und SETLST.COM stehen dem JOYCE-Anwender zur Verfügung. Um Änderungen der Hardware-Einstellungen vorzunehmen, auch kann man zum Beispiel den BASIC-Interpreter laden und über PRINT/LPRINT-Kommandos direkt eine Einstellung ändern. Das setzt jedoch immer voraus, daß diese Programme jeweils sofort verfügbar sind, und damit belegen sie wieder "kostbaren" Speicherplatz auf den Start- und Arbeitsdisketten. Ein Diskettenwechsel, nur um den Zeichensatz des Bildschirms umzuschalten, wird wohl in den wenigsten Fällen durchgeführt, da akzeptiert man schon eher fremdsprachige Zeichen.

Es gibt jedoch auch alternative Möglichkeiten, solche Änderungen vorzunehmen, zum Beispiel direkt von der Tastatur. Leider sind diese Möglichkeiten nicht in den Handbüchern dokumentiert, so daß viele Benutzer des JOYCE bisher darauf verzichten mußten.

Ein Hinweis: Die nun folgenden Ausführungen beziehen sich auf die Systemebene von CP/M, das heißt, das CP/M-Programm ist geladen, und der System-Prompt 'A>' ('B>' oder 'M>') wird angezeigt.

Falls Sie Ihre Tastatur umdefiniert haben, sei es mit dem Dienstprogramm SETKEYS.COM oder mit einem ähnlichen Programm, ist es natürlich erforderlich, daß die im folgenden beschriebenen Tasten-Codes anderweitig erzeugt werden können, da Sie sonst auf die 'alternative Steuerung' von Bildschirm und Drucker verzichten müßten.

Der Bildschirm...

Funktionen des Bildschirms, wie das Löschen, das Bewegen des Cursors oder das Definieren eines Darstellungsfeldes, werden über Steuercodes beziehungsweise Steuercode-Sequenzen aufgerufen, die den Bildschirmtreiber veranlassen, die jeweils gewünschte Operation durchzuführen. Der Begriff 'Bildschirmtreiber' bezieht sich hierbei auf jenen Teil des Systemprogramms, der für die Ausgabe der Zeichen an den Bildschirm zuständig ist. Die möglichen Steuercode-Sequenzen und deren Funktionen sind im Benutzerhandbuch Teil 2, Anhang 3 ausführlich erläutert, daher wollen wir an dieser Stelle auf eine nähere Beschreibung verzichten.

Diese Zeichen und Steuercodes werden normalerweise vom Anwenderprogramm über die normalen Ausgaberoutinen ans Systemprogramm übergeben, in BASIC zum Beispiel mit dem 'PRINT'-Kommando. Auf Systemebene, wo der oben erwähnte CCP das Sagen hat, scheint es allerdings nicht so einfach, diese Steuercode-Sequenzen zu erzeugen. Die Eingabe von

`PRINT CHR$(27)+"E"`,

dem Kommando, mit dem unter BASIC der Bildschirm gelöscht werden kann, weist der CCP mit "freundlichem" Fragezeichen zurück. Mit einem solchen Kommando kann er nichts anfangen, es sei denn, ein Programm mit Namen 'PRINT.COM' existiert und ist in der Lage, die Eingabe zu verarbeiten.

CLS in CP/M?

Die Erfahrung im Umgang mit dem JOYCE-System zeigt jedoch, daß man bestimmte Steuercodes auch direkt von der Tastatur aus erzeugen kann. Durch Drücken der Tasten

>EXIT<, >H<, >EXIT<, >E<, >RETURN<

wird zum Beispiel der Bildschirm gelöscht und der Cursor in die linke obere Ecke gesetzt. Auf diese Weise kann, falls notwendig, die gerade angezeigte Information vor unbefugten Blicken geschützt werden. Dafür, daß diese Tastenfolgen die entsprechende Wirkung

zeigen, zeichnen hauptsächlich zwei Faktoren verantwortlich. Zunächst liegt es an der Voreinstellung der Tastatur, die der 'EXIT'-Taste unter CP/M den ASCII-Wert 27 (&H1B) zuordnet. Dieser ASCII-Wert leitet für gewöhnlich die oben schon erwähnten Steuercode-Sequenzen ein und bietet somit den ersten Anhaltspunkt für eine alternative Steuerung der Monitor-Funktionen. Allerdings, wenn Sie die 'EXIT'-Taste drücken, erscheint dabei auf dem Bildschirm nicht wie erwartet ein dem ASCII-Code entsprechendes Zeichen, sondern zwei Zeichen, '↑' und 'Ä'. Bei der Übersetzung des Tasten-Codes stellt das CP/M-Programm nämlich fest, daß der Taste ein sogenanntes Erweiterungs-String (Expansions-String) zugeordnet ist, das heißt, diese Taste erzeugt mehrere Zeichen mit einem einzigen Tastendruck. In diesem Fall wird der Control-Code '↑Ä' erzeugt, der natürlich auch durch gleichzeitiges Drücken der Tasten >ALT< + >Ä< erzeugt werden kann. Eine einzelne Taste läßt sich jedoch in den meisten Fällen einfacher bedienen als zwei Tasten gleichzeitig: Durch die Übersetzung der 'EXIT'-Taste wird also nicht der Steuercode 27 an den Bildschirm geschickt, sondern die ASCII-Codes für die Zeichen '↑' und 'Ä'. Erst die Tatsache, daß der CCP die Eingaben, die er nicht verarbeiten kann, wieder dem Nutzer am Bildschirm anzeigt (versehen mit dem "freundlichen" Fragezeichen), ermöglicht die Interpretation als Steuerzeichen (auch ESCape-Code). Alle Zeichen einer Eingabezeile für den CCP werden in einem Zeichenpuffer aufbewahrt, und zwar nicht die übersetzten Zeichen, sondern die von der Tastatur erzeugten. Kann der CCP die Eingabe nicht verarbeiten, werden die Zeichen aus diesem Puffer wieder an den Bildschirm geschickt, versehen mit einem Fragezeichen, um so dem Anwender zu zeigen, daß etwas nicht in Ordnung ist. Bei dieser Ausgabe werden die einzelnen Zeichen jedoch nicht noch einmal übersetzt, so daß hier der ASCII-Code 27 (anstelle von '↑Ä') an den Bildschirmtreiber übergeben wird. Wenn wir nun dafür sorgen, daß diesem ESC-Zeichen die entsprechenden Steuer-Zeichen folgen (s.o.), dann stellt unsere Eingabe für den Bildschirm eine Steuercode-Sequenz dar, die eine bestimmte Funktion erfüllen soll. Man kann also auch mit der Tastatur solche Sequenzen erzeugen, indem man sich die "Eigenarten" des CCP zunutze macht.

```
Tastensequenz:                       Funktion:

'EXIT' + '0'                         Statuszeile aus
'EXIT' + '1'                         Statuszeile ein
'EXIT' + '2' + ...                   Zeichensatz auswählen
               '§'                   ZS 0 (Amerikanisch)
               'A'                   ZS 1 (Französisch)
               'B'                   ZS 2 (Deutsch)
               'C'                   ZS 3 (Englisch)
               'D'                   ZS 4 (Dänisch)
               'E'                   ZS 5 (Schwedisch)
               'F'                   ZS 6 (Italienisch)
               'G'                   ZS 7 (Spanisch)

 'EXIT' + ...
      'A', 'B', 'C', 'D', 'E',       entsprechen den im Handbuch
      'H', 'I', 'J', 'K', 'L',       aufgeführten Escape-Sequenzen
      'M' oder 'N'                   (ESC ...)

 'EXIT' + 'X' + oz + ls + h + b   definiert ein Darstellungsfeld

 Die Werte für oz (obere Zeile), ls (linke Spalte), h (Höhe) und
 b (Breite) können mit den folgenden Tasten eingegeben werden:

  0 = '^'*          16 = '0'          32 = '§'          48 = 'P'
  1 = '!'**         17 = '1'          33 = 'A'          49 = 'Q'
  2 = '"'           18 = '2'          34 = 'B'          50 = 'R'
  3 = '#'           19 = '3'          35 = 'C'          51 = 'S'
  4 = '$'           20 = '4'          36 = 'D'          52 = 'T'
  5 = '%'           21 = '5'          37 = 'E'          53 = 'U'
  6 = '&'           22 = '6'          38 = 'F'          54 = 'V'
  7 = '''           23 = '7'          39 = 'G'          55 = 'W'
  8 = '('           24 = '8'          40 = 'H'          56 = 'X'
  9 = ')'           25 = '9'          41 = 'I'          57 = 'Y'
 10 = '*'           26 = ':'          42 = 'J'          58 = 'Z'
 11 = '+'           27 = ';'          43 = 'K'          60 = 'Ö'
 12 = ','           28 = '<'          44 = 'L'          61 = 'Ü'
 13 = '-'           29 = '='          45 = 'M'          62 = '^'
 14 = '.'           30 = '>'          46 = 'N'          63 = '_'
 15 = '/'           31 = '?'          47 = 'O'          89 = 'ā'***

 *   Pfeil nach oben (Cursor-Taste) oder >ALT< + >_<
 **  dieses Zeichen muß 2-fach eingegeben werden
 *** wird durch die Tastenkombination 'EXTRA'+'1' erzeugt

'EXIT' + 'Y' + z + s                 setzt den Cursor an die
                                     Position z(eile), s(palte)
```

Abbildung 1: Tastensequenzen zur direkten Steuerung der Bildschirm-Funktionen

Wollen Sie beispielsweise den amerikanischen Zeichensatz benutzen, genügt die Tastenfolge

>EXIT<, >2<, >@<, >RETURN<,

um die interne Umschaltung zu aktivieren. Die 'RETURN'-Taste muß übrigens immer betätigt werden, um die Eingabe abzuschließen und an den CCP zu übergeben. Leider lassen sich nicht alle Steuercodes auf diese Art erzeugen, denn der CCP hat noch eine weitere Eigenart, die wir beachten müssen. Bevor die Zeile erneut ausgegeben wird, werden alle Eingaben in Großbuchstaben umgewandelt. Das bedeutet für uns: Es sind nur solche Steuercode-Sequenzen möglich, die sich mit Großbuchstaben beziehungsweise Ziffern realisieren lassen.

Tabelle 1 gibt Auskunft, welche Sequenzen Sie hier benutzen können. Ähnliche Einschränkungen ergeben sich bei den Zahlenwerten für die Größe eines Darstellungsfeldes und die Positionierung des Cursors auf dem Bildschirm. Es lassen sich nicht alle Größen und Positionen eingeben, da der CCP keine ASCII-Zeichen im Bereich von &H61 bis &H7B (= Kleinbuchstaben) ans Systemprogramm übergibt. Auch hier finden Sie in Tabelle 1 die entsprechenden Werte und die dazu gehörenden Tasten. Der im Handbuch vorgeschriebene Offset von &H20 braucht bei der Eingabe der Zahlenwerte nicht mehr beachtet werden. Bei den in Tabelle 1 aufgeführten Werten ist dieser Offset schon eingeschlossen.

... und der Drucker

Da auch der Drucker des JOYCE über Steuercodes kontrolliert werden kann, liegt es nahe, hier ebenfalls die Tastatur zur direkten Steuerung einzusetzen. Die benötigten ESCAPE-Sequenzen (vgl. Benutzerhandbuch Teil 2, Anhang 2) müssen nur wieder in die entsprechenden Tastenfolgen übersetzt

Tastensequenz:	Funktion:
'EXIT' + '§'	RESET
'EXIT' + '0'* 'EXIT' + '1'* 'EXIT' + '2'* 'EXIT' + '3' + n 'EXIT' + 'A' + n*	Zeilenabstand auf 8 Zeilen/Zoll Zeilenabstand auf 10 Zeilen/Zoll Zeilenabstand auf 6 Zeilen/Zoll Zeilenabstand auf n/216 Zoll Zeilenabstand auf n/ 72 Zoll
'EXIT' + 'J' + n*	Papiervorschub n/216 Zoll
'EXIT' + 'C' + n* 'EXIT' + 'C' + '0' + n*	Seitenlänge n Zeilen Seitenlänge n Zoll
'EXIT' + 'N' + n* 'EXIT' + 'O'	n Zeilen am Seitenende freilassen keine Zeilen am Seitenende freilassen
'EXIT' + '8' 'EXIT' + '9'	Papierende ignorieren Papierende anzeigen
'EXIT' + '!' + '!' + '§' 'EXIT' + '!' + '!' + 'A' 'EXIT' + '!' + '!' + 'D'	Schriftart PICA (10 cpi) Schriftart ELITE (12 cpi) Schriftart KOMPAKT (17 cpi)
'EXIT' + 'E'* 'EXIT' + 'F'	Fettdruck ein Fettdruck aus
'EXIT' + 'G' 'EXIT' + 'H'*	Doppeldruck ein Doppeldruck aus
'EXIT' + 'W' + '1' 'EXIT' + 'W' + '0'	Breitschrift ein Breitschrift aus
'EXIT' + '4' 'EXIT' + '5'	Kursiv-Schrift ein Kursiv-Schrift aus
'EXIT' + 'S' + '0' 'EXIT' + 'S' + '1' 'EXIT' + 'T'	Hochstellen (Superscript) Tiefstellen (Subscript) Hoch-/Tiefstellen aus
* Diese Tastensequenzen haben auch Einfluß auf die Bildschirmsteuerung	
Die Werte für n sind entsprechend aus Abbildung 3 zu wählen	

Abbildung 2: Tastensequenzen zur direkten Steuerung der Drucker-Funktionen

und dann an den Druckertreiber übergeben werden. Dabei taucht gleich die Frage auf: Wie bringt man den CCP dazu, die erzeugten Steuercode-Sequenzen an den Drucker zu schicken? Aber hier haben die Programmierer von CP/M mitgedacht und eine Protokoll-Funktion für den Drucker vorgesehen. Mit dem Control-Code '↑P' kann die Zeichenausgabe parallel zum Bildschirm auch auf den Drucker geleitet werden, das heißt, alles, was auf dem Bildschirm erscheint, wird, wenn die Protokoll-Funktion eingeschaltet ist, auch an den Drucker geschickt.

Das Zeichen '↑P' läßt sich, sofern Sie Ihre Funktionstasten nicht neu belegt haben, mit >f7/f8< oder durch gleichzeitiges Drücken der Tasten >ALT< + >P< erzeugen.

Ein- und Ausschalten der Protokoll-Funktion geschieht mit dem gleichen Zeichen (beziehungsweise Tasten), beim ersten Mal ertönt ein akustisches Signal, und die Protokoll-Funktion ist eingeschaltet, mit dem zweiten Betätigen der Taste(n) wird die Protokoll-Funktion wieder abgeschaltet (kein akustisches Signal).

Aber es gibt noch einiges zu beachten, bevor Sie nun die gewünschten Steuercodes an den Drucker übertragen können. Zunächst einmal ist es notwendig, die Protokoll-Funktion für den Drucker erst am Ende der Eingabezeile einzuschalten, denn sonst werden zunächst anstelle des ESC-Codes die Zeichen '↑' und 'Ä' an den Drucker übertragen. Damit kann aber in der Regel keine Umschaltung der Drucker-Funktionen durchgeführt werden.

Eine weitere Einschränkung bildet, wie auch schon bei der Bildschirm-Steuerung, die Eigenart des CCPs, alle Kleinbuchstaben in große umzuwandeln. Dadurch wird die Anzahl der für uns verfügbaren Steuercode-Sequenzen wie beim Monitor vermindert.

3 = 'ALT' + 'C' 4 = 'ALT' + 'D' 12 = 'ALT' + 'L' 14 = 'ALT' + 'N' 15 = 'ALT' + 'O' 17 = 'ALT' + 'Q' 19 = 'ALT' + 'S' 20 = 'ALT' + 'T' 22 = 'ALT' + 'V' 25 = 'ALT' + 'Y' 26 = 'ALT' + 'Z' 27 = 'ALT' + 'Ä' 28 = 'ALT' + 'Ö' 29 = 'ALT' + 'Ü' 30 = 'ALT' + '^' 31 = 'ALT' + '_'	Die Werte > 33 entsprechen den ASCII-Werten der Zeichen gemäß Handbuch Teil 2, Anhang 1 und werden mit den entspr. Tasten erzeugt. Beispiel: 'EXIT' + 'J' + '¿' (= 'ALT'+'1') löst einen Papiervorschub von 182/216 Zoll aus

Abbildung 3: Tastenkombinationen zur Eingabe von Zahlenwerten (n) für die Druckersteuerung

In Tabelle 2 finden Sie die wichtigsten Tasten- beziehungsweise Steuercode-Sequenzen aufgeführt. Ebenso müssen wieder für eventuell benötigte Zahlenwerte (zum Beispiel Zeilenabstand, Seitenlänge, etc.) Einschränkungen in Kauf genommen werden. Informationen hierzu finden Sie in Tabelle 3 aufgeführt. Hier fällt besonders ins Gewicht, daß eine Zeichensatzumschaltung wie für den Bildschirm nicht möglich ist. Die benötigten Zahlenwerte werden vom CCP nicht akzeptiert, beziehungsweise sie lassen sich mit der Tastatur nicht erzeugen. Auch könnte das vom CCP erzeugte Fragezeichen bei der Druckerausgabe stören, es läßt sich aber leider nicht vermeiden, denn wie oben schon ausgeführt, zeigt der CCP damit an, daß er die Eingabe nicht verarbeiten kann. Damit hat man aber dann zum Beispiel bei der Änderung der Schriftart gleich eine Anzeige, ob die Umschaltung erfolgreich durchgeführt wurde.

Ein Beispiel für die Erzeugung einer Steuersequenz für den Drucker: Drücken Sie nacheinander folgende Tasten
>EXIT<, >!<, >!<, >D<, >f7<, >RETURN<, >f7<
Diese Sequenz stellt den Drucker auf die Schriftart KOMPAKT (17 cpi) ein. Das Zeichen '!' muß für die Steuer-Sequenzen stets doppelt eingegeben werden, da der CCP es sonst als Kommando-Trennzeichen betrachtet und es bei der Verarbeitung entfernt.

Einige der für den Drucker benötigten Steuercode-Sequenzen haben auch Einfluß auf den Monitor. Da diese Sequenzen primär an den Bildschirm geschickt und nur über die Protokoll-Funktion an den Drucker gelangen, können sie also auch die Bildschirmausgabe beinflussen. Dies sollten Sie bei der Anwendung beachten.

(Norbert Finke/rs)

Compilervergleich

Wie leistungsfähig sind PD-Compiler?

Da auch für den PCW viele Programme aus der Public Domain verfügbar sind, stellt sich die Frage, wie gut oder schlecht diese sind, und ob man für einen so niedrigen Preis wirklich gute Qualität erwarten kann. Aus diesem Grund wurden einige Compiler aus der Public Domain 'offiziellen' Programmiersprachen gegenübergestellt.

Für diesen Vergleich wurde die Berechnung der alternierenden Formel (1-1/2+1/3-1/4+ … +1/9999-1/10000) herangezogen. Hierbei erfolgte die Berechnung auf vier verschiedenen Wegen:

- der Reihe nach: von links nach rechts.
- die Abarbeitung erfolgt von links nach rechts; es werden zunächst alle positiven und negativen Glieder berechnet, dann wird subtrahiert.
- der Reihe nach: von rechts nach links.
- die Abarbeitung erfolgt von rechts nach links; es werden zunächst alle positiven und negativen Glieder berechnet, dann wird subtrahiert.

Das exakte Ergebnis mit 21 Stellen hinter dem Komma lautet:

0.693 097 183 059 945 296 917

Bei dem folgenden Vergleich wurde deshalb die Rechengenauigkeit der jeweiligen Compiler beziehungsweise Interpreter ermittelt. Ferner wurde die Zeitdauer für die jeweilige Berechnung festgestellt.
Folgende Interpreter und Compiler wurden in die Untersuchung einbezogen:

- ○ Mallard BASIC (einfache Genauigkeit)
- ○ Mallard BASIC (doppelte Genauigkeit)
- ○ Turbo Pascal (Borland/Heimsoeth)
- ○ JRT Pascal (PD SIG/M #82 oder deutsche Version von Kotulla, Nürnberg)
- ○ Small C (deutsche Version von Kotulla, Nürnberg)
- ○ E-BASIC (deutsche Version von Kotulla, Nürnberg)
- ○ ALGOLM (PD CPMUG #328)

Wer rechnet am genauesten

Für jeden der genannten Compiler wurde ein Programm erstellt, das diese Berechnung ausführte. Die Ergebnisse der Berechnung können Sie Abbildung 1 entnehmen.
Wie Sie feststellen werden, liefert sowohl das JRT-Pascal-Programm als auch das Small-C-Programm mit mindestens zehn Stellen nach dem Komma die besten Ergebnisse.
Das Mittelfeld bilden gemeinsam Mallard BASIC bei einfacher Genauigkeit und E-BASIC. Beide liefern Ergebnisse, bei denen nur die ersten drei Stellen hinter dem Komma noch stimmen. Bei doppelter Genauigkeit dagegen bringt es Mallard BASIC auf immerhin neun Stellen hinter dem Komma – noch vor Turbo Pascal, bei dem immerhin noch die ersten acht Stellen nach dem Komma stimmen. ALGOLM liefert das schlechteste Ergebnis: nur zwei Stellen hinter dem Komma stimmen.

E-BASIC bietet dem Programmierer keine Möglichkeit, zwischen einfacher und doppelter Genauigkeit zu unterscheiden. Hier wird grundsätzlich mit einfacher Genauigkeit gearbeitet.

Dieses Experiment zeigt zum einen, daß Mallard BASIC gar nicht so schlecht ist, zum anderen aber auch, daß die PD-Compiler JRT Pascal und Small C ganz respektable Ergebnisse vorweisen können, während E-BASIC und ALGOLM nur mit äußerst bescheidenen Ergebnissen aufwarten können.

Wer ist der Schnellste

Da nicht nur die Rechengenauigkeit, sondern auch die Zeit interessant ist, die der Compiler für die Berechnung benötigt, müßte dieser Faktor auch mit einbezogen werden (Meßergebnisse in Abbildung 2).

Es ist schon erstaunlich, wie unterschiedlich die Ergebnisse sind. Vor allem enttäuscht das ALGOLM-Programm mit einer Laufzeit von mehr als einer Stunde mit einem Rechenergebnis, bei dem nur die ersten zwei Stellen nach dem Komma stimmen.

Trotzdem ist der ALGOLM-Compiler sehr wohl geeignet sich in die Programmiersprache ALGOL einzuarbeiten. Nur bei der Massendatenverarbeitung hat der Compiler seine Grenzen.

Compiler:	Ergebnis 1	Ergebnis 2	Ergebnis 3	Ergebnis 4
MBASIC (einf. Gen.)	0.6930944	0.6931019	0.6930972	0.6930971
MBASIC (dopp. Gen.)	0.6930971832625801	0.6930971832625801	0.6930971832625801	0.6930971832625801
Turbo Pascal	0.6930971830200	0.6930971805300	0.6930971830600	0.6930971830000
JRT Pascal	0.6930971830099	0.6930971830530	0.6930971830598	0.6930971830588
Small C	0.6930971830479	0.6939971830624	0.6930971830624	0.6930971830370
E-BASIC	0.693094	0.693102	0.693097	0.693097
ALGOLM	0.6931382565	0.6931583015	0.6926877103	0.6925588429

Abbildung 1: Die Ergebnisse lassen erhebliche Abweichungen erkennen

Compiler	Rechenzeit in Sekunden
Mallard BASIC (einf. Gen.)	158
Mallard BASIC (dopp. Gen.)	201
Turbo Pascal	103
JRT Pascal	716
Small C	672
E-BASIC	251
ALGOLM	über eine Stunde

Abbildung 2: In Sachen Geschwindigkeit kann sich nur – wie erwartet – Turbo Pascal gegen Mallard BASIC behaupten

Gar nicht so schlecht

Mallard BASIC bringt auch hinsichtlich seiner Geschwindigkeit ein beachtliches Ergebnis. Der Name Mallard ist sicherlich zu recht gewählt worden ("Mallard" ist eine Dampflokomotive aus dem Jahr 1938, die bis heute den Geschwindigkeitsweltrekord hält). Es ist eigentlich bedauerlich, daß diese und weitere tolle BASIC-Eigenschaften (beispielsweise JETSAM) auf dem PCW viel zu wenig bekannt sind.

Die Spitzenposition von Turbo Pascal war zu erwarten. Wer allerdings eine erhöhte Rechengenauigkeit benötigt, ist mit JRT Pascal oder Small C ganz gut bedient und muß nicht zu erheblich teureren, kommerziell vertriebenen Compilern greifen.

Fazit

Die vorliegenden Ergebnisse unterstreichen, daß einige der in der Public Domain für den Joyce verfügbaren Compiler nicht nur für Schulungs- und Testzwecke eingesetzt werden können, sondern durchaus eine brauchbare Alternative zu den erheblich teureren Handelsprodukten darstellen.

(Wolfgang Träber/rs)

Damit Sie den Test eventuell selber noch einmal durchführen können, haben wir auf der Databox die Testprogramme für die jeweiligen Sprachen kopiert.

MINI-DOS-Toolbox

Eine Utility-Sammlung für JOYCE

In der Ausgabe 10/89 der PCI wurde bereits MFU (Multi Format Utility) vorgestellt, mit dem der Joyce weitgehend kommunikationsfähig wird. Mit dem heute vorzustellenden MINI-DOS wird unser zuverlässiger Kamerad ebenfalls befähigt, MS-DOS-Disketten zu lesen, sie zu formatieren, Umlaute nach MS-DOS zu konvertieren und von CP/M nach MS-DOS und umgekehrt zu kopieren. All das läuft leicht verständlich in deutscher Sprache und menügesteuert ab.

Es stellt sich natürlich die Frage, wofür dererlei Konvertierungen überhaupt zu gebrauchen sind.
Nun, wenn zum Beispiel Autoren ihre Manuskripte auf einer 3,5- oder 5,25-Zoll-Diskette, bereits MS-DOS-konvertiert, verschicken, werden Redaktionen und Verlage sicherlich dankbar und eher bereit sein, das Manuskript anzunehmen.
Ferner ist es so möglich, zum Beispiel Texte, die während der Arbeit auf dem PC nicht ganz fertig geworden sind, zu Hause auf dem Joyce weiterzubearbeiten und am nächsten Tag wieder mitzubringen.
Das Programmpaket MINI-DOS besteht aus den Files:

```
MD04.COM (25K)
UK09.COM (18K)
PF07.COM ( 3K)
```

Zusätzlich befinden sich noch ausführliche Anweisungen in ASCII (*.ASC für LOCOSCRIPT) und PROWORT (*.PRW) auf der Diskette. Der Aufruf aller COM-Dateien erfolgt bekanntlich aus dem System heraus (aus PROWORT mit vorgesetztem Sternchen).

Disketten formatieren

Nach dem Aufruf von PF07 meldet sich das Programm mit einer Ansage und der Abfrage: 'Wählen Sie ein Laufwerk (A oder B):'
Nachdem die Wahl getroffen wurde, muß das Diskettenformat angegeben werden. Bei 5,25-Zoll-Disketten sind das in der Regel 360 kByte (40 Spuren), bei 3,5-Zoll-Laufwerken 720 kByte (80 Spuren). Bequem ist, daß PF07 keine Umschaltung des Zweitlaufwerks (40 oder 80 Spuren) verlangt. Das Programm macht alles richtig. Damit nicht aus Versehen eine falsche, eventell mit wichtigen Texten vollgestopfte Diskette formatiert und damit gelöscht wird, fragt PF07 noch ab, ob die Formatierung wirklich durchgeführt werden soll und beginnt erst mit der Arbeit, wenn der ausdrückliche Befehl dazu erfolgt. Geht dann trotzdem etwas schief, waren weder Programm, noch Computer ursächlich daran beteiligt.
Wird die Frage: 'Wollen Sie wirklich...?' verneint, steigt das Programm automatisch aus. Es kann übrigens nicht direkt unterbrochen werden, was bei einem versehentlichen Aufruf ärgerlich ist.

Konvertieren

Wird der Umlautkonverter UK09 aufgerufen, fragt das Programm zunächst nach dem Dateinamen. Diesen sollte man sich vorher notieren, denn aus dem Programm heraus kann nicht mehr ins Verzeichnis geschaut werden. Außerdem sollte tunlichst eine Kopie der Datei behandelt werden, denn der Umlautkonverter legt keine BAK-Datei an und formt das angegebene File restlos um.
Nach der Namenseingabe erscheint folgendes Menü:

```
Suchen Umlaut-Zeichensatz
(1) = Prowort Version 2.xx
(2) = Locoscript Version 2.xx
(3) = Locoskript Version 1.xx
(4) = MS-DOS Version X.xx
```

Aha, auch die MS-DOS-Zeichen lassen sich Joyce- und CPC-verständlich umwandeln!
Nach der richtigen Wahl erscheint das gleiche Menü noch einmal, nur mit der Überschrift:

```
'Ersetzen durch Umlaut-Zeichensatz:'
```

Siehe da: UK09 läßt sich auch dazu verwenden, um Prowort-Dateien problemlos nach LocoScript und umgekehrt zu konvertieren. Das geht einfacher als mit dem PROWORT-Programm KONVERT.
Leider steigt UK09 nach der Konvertierung gleich aus und muß zu einer weiteren Umwandlung neu geladen werden. Hier wäre eine Abfrage nach weiteren Konvertierungswünschen zu begrüßen.

Klein, aber oho

Nun kommen wir zum eigentlichen Hauptprogramm. Nach dem Aufruf von MD04 will MINI-DOS zunächst wissen, in welchem Laufwerk sich die MS-DOS-Diskette befindet und wo die CP/M-Diskette steckt.
Im anschließend eingeblendeten Menü wird angezeigt, welches Laufwerk, entsprechend der Eingabe, auf CP/M und welches nun auf MS-DOS getrimmt ist. Die aufgezeigten Optionen lauten:

```
(1) Datei-Transfer CP/M ==> MS-DOS
(2) Datei-Transfer MS-DOS ==> CP/M
(3) Directory CP/M Diskette
(4) Directory MS-DOS Diskette
(5) Datei loeschen CP/M Diskette
(6) Datei loeschen MS-DOS Diskette
(7) FAT-Ausgabe MS-DOS Diskette
(8) FAT reparieren MS-DOS Diskette
(9) Exit - Programm verlassen
```

Zu den Optionen 1 bis 6 ist nicht viel zu sagen. Allerdings funktioniert der Datentransfer MS-DOS ==> CP/M nicht immer. Joyce zeigt gelegentlich zwar den Inhalt des Directories richtig an, behauptet dann aber, es wären keine Daten auf den einzelnen Spuren der Disk zu finden. Die Ursache ist im Fremdlaufwerk zu suchen. Zwischen den Laufwerken verschiedener Hersteller kommt es mangels einheitlicher Konfiguration schon mal zu Mißverständnissen. In solchen Fällen bleibt als einziger Ausweg, die problematische Diskette auf einem fremden Gerät zu kopieren, dessen Laufwerkskonfiguration zu der des eigenen kompatibler ist.
Bleibt noch die Option FAT-Ausgabe. Mit Fett hat das nichts zu tun. In der File Allocation Table ist die Belegung der Diskette gespeichert. FAT REPARIEREN bedeutet im Prinzip nichts anders, als eine defekte Datei aus einer Reservedatei zu erneuern. Wird das nötig, ist zusätzlich ein Beruhigungsmittel erforderlich (für den Anwender natürlich!).

Eine runde Sache

Die Gesamtbeurteilung von MINI-DOS fällt positiv aus. Dem Anwender wird für wenig Geld ein nützliches Programmpaket geboten, das sowohl Joyce als auch den CPC 6128 den PCs etwas annähert.

(Bruno Melchert/rs)

Sollten Sie Interesse an diesem Programm haben, so wenden Sie sich bitte an den Soft- und Hardwareversand, Ulrike Becker, Fasanenweg 2, 6690 St. Wendel 8. Hier können Sie das Programm zu einem Preis von 50,– DM beziehen.

Basic klar und deutlich

Übersichtliche Programmierung in Mallard-Basic

Basic ist an sich eine recht gute und leicht erlernbare Programmiersprache. Leider hat es sich bei den meisten Programmierern eingebürgert, ihre Listings so unübersichtlich zu schreiben, daß es demjenigen, der ein solches Programm abtippen oder gar umschreiben will, sehr schwerfällt, die einzelnen Programmabläufe genau nachzuvollziehen.

```
1     REM ************************
2     REM * Demonstrationsprogramm *
3     REM ************************
4     REM
9     ' Hauptprogramm
10    GOSUB 10000 'Grafikinitialisierung
20    GOSUB 10100 'Variablenzuweisung
30    PRINT cls$
40    RESTORE 1000
50    FOR a=1 TO 10
60     READ a$
70     PRINT FNlocate$(30,a*2+5);a$
80    NEXT a
90    IF INKEY$='' THEN GOTO 90 ELSE wahl=val(INKEY$)
100   ON WAHL GOTO 1500,2000,2500,3000,3500,4000,4500,5000,5500,6000
999   ' Datas für Hauptmenü
1000 DATA Menüpunkt 1,Menüpunkt 2,Menüpunkt 3,Menüpunkt 4,Menüpunkt 5
1010 DATA Menüpunkt 6,Menüpunkt 7,Menüpunkt 8,Menüpunkt 9,Menüpunkt 10
1499 ' Menüpunkt 1
5999 ' Menüpunkt 10
9999 ' Grafikinitialisierung
10000 DEF FNlocate$(x,y)=CHR$(27)+"Y"+CHR$(32+y)+CHR$(32+x)
10010 DEF FNwindow$(z,s,h,b)=CHR$(27)+"X"+CHR$(z+32)+CHR$(s+32)+CHR$(b+31)+CHR$(h+31)
10090 RETURN
10098 'Variablenzuweisung
10099 'Umrandung
10100 ou$=CHR$(154):ol$=CHR$(150):OR=CHR$(156)
10120 lr$=CHR$(149):ul$=CHR$(147):ur$=CHR$(153)
10199 'Sonstiges
Ok
                                                        Laufwerk ist A:
```

Hier haben es diejenigen schon besser, die in Turbo-Pascal oder anderen Hochsprachen programmieren, da es hier ganz besonders wichtig ist, auf die Struktur der Programme zu achten, da sonst durch wilde Aneinanderkettung von Prozeduren, Funktionen und Variablendeklarationen die Programme unleserlich werden.
Auch in Basic gibt es einige Regeln, welche der Übersichtlichkeit halber eingehalten werden sollten.

Grundgedanken

Vor dem Anschalten des Computers und dem Starten des Programmierens sowie mitten in der Programmierung ist Schreibarbeit angesagt. Grundüberlegungen über die Struktur und Aufteilung der Programme sollten auf Papier gebracht werden.
Überlegungen, wie zum Beispiel, an welche Stelle ein bestimmtes Unterprogramm gesetzt werden soll, können natürlich erst während des Eingebens in das bereits bestehende Programmschema eingebaut werden.
Dieses Schema sollte neben den einzelnen Unterprogrammen auch noch Hinweise über Art, Anfangszeilennummer, eventuell Länge und über die benutzten Variablen enthalten.

Zeilenaufteilung

Bei der Gestaltung des Programms sollten einige Punkte beachtet werden:

● Grundsätzlich kann davon ausgegangen werden, daß ein Programm aus mehreren Teilen besteht. Als erstes wäre da das Hauptprogramm zu nennen, welches je nach Bedarf mit den Befehlen GOTO oder GOSUB Unterprogramme aufruft, in welchen zum Beispiel Berechnungen stehen, die während des Programmablaufs des öfteren benötigt werden. Hier gilt, daß das Hauptprogramm am Anfang und die Unterprogramme am Ende stehen, da der Basic-Interpreter, bevor er die Befehle in einer Zeile ausführt, immer von der ersten bis zu dieser Zeile durchzählt. Also müssen auch Unterprogramme, welche häufiger benötigt werden, weiter vorne stehen als diejenigen, welche kaum benötigt werden. Die Anordnung der einzelnen Programmteile hat also auch eine Auswirkung auf die Geschwindigkeit.

● Zwischen den einzelnen Programmteilen sollten regelmäßige Abstände eingehalten werden. Je nach Länge bieten sich 100er oder 500er bei Unterprogrammen und 1000er oder 10000er Zeilensprünge zwischen dem Haupt- und dem ersten Unterprogramm an. Die Aufteilung könnte dann ungefähr so aussehen:

10.. 9990 Hauptprogramm
10000..10090 Unterprogramm 1
10100..10190 Unterprogramm 2
...

Hilfe mit REMarks

Um diese Einteilung nicht durcheinanderzubringen, sollten Sie während der Programmierung tunlichst vermeiden, mit dem Befehl RENUM zu arbeiten, da sonst die ganze Übersicht wieder verlorengeht.

● Die Anfänge von Unterprogrammen sollten mittels des REMark-Befehls mit einem kurzen Kommentar versehen werden, damit derjenige, der dieses Programm später einmal in die Hände bekommt, ohne große Schwierigkeiten eigene Änderungen anbringen kann.

● Als Unterprogramme sollten nur Funktionen und Berechnungen benutzt werden, die mehrmals im Programm wiederholt werden. Sollte dies nicht der Fall sein, der Programmblock innerhalb des Hauptprogramms aber eine besonders wichtige Aufgabe haben, gibt es ja noch die Möglichkeit, auch hier mit REMarks zu arbeiten.

● Wenn innerhalb des Programms ein GOTO oder GOSUB-Befehl zum Ansprung von Unterprogrammen verwendet wird, sollte direkt dahinter mittels REMark gekennzeichet werden, wohin dieser Sprung erfolgt. So läßt sich jeder Schritt genau überprüfen, ohne erst die entsprechende Zeile aufzulisten.

Anschauliches

Um die Übersicht innerhalb des Haupt- oder Unterprogramms zu erhöhen, kann man auch so einiges tun.

● Bei FOR TO NEXT- und WHILE WEND-Schleifen ist es angebracht, die Zeilen, welche zwischen FOR beziehungsweise WHILE und NEXT/WEND stehen, um ein oder zwei Zeichen nach rechts einzurücken.

● Innerhalb einer Zeile sollten nicht allzuviele Befehle stehen. Wenn jedoch mehrere Befehle verwandt werden, ist es ratsam, nur zusammengehörige Befehle zu schreiben (wie zum Beispiel 100 RESTORE 20000: FOR n=0 TO 100:READ a$:a=VAL("&H"+a$): POKE adr+n,a:NEXTn).

(rs)

Gamers Message

Auch diesen Monat geht es wieder auf die Reise in die Welt der Phantasie. Mythen und Mysterien erkunden, Wichte und Getüme bekämpfen, Prinzessinnen und andere holde Geschöpfe befreien – kann man seine Freizeit besser verbringen?
Unser Erste-Hilfe-Paket befaßt sich dieses Mal unter anderem mit...

Myth – ein Mythos wird entschleiert

Nach einem einjährigen Gelage im Olymp, bei dem Obergott Zeus feststellt, daß das alles so nicht weitergehen kann, findet sich Poseidon vor den Toren des Hades Auge in Auge mit der neunköpfigen Hydra wieder. Seiner göttlichen Fähigkeiten beraubt, bleibt ihm nur sein treuer Dreizack und viel Phantasie, um die ihm gestellte Aufgabe zu lösen: die Kopfbedeckung des Gottes der Unterwelt zu klauen. Nur mit Hades' Hut in der Hand kann er wieder zur Party im Götterhimmel zurückkehren.
An der Hydra ist leider vorerst kein Vorbeikommen, da der Dreizack noch ziemlich wirkungslos von ihr abprallt. Auch im Sumpf geht's zunächst nicht weiter. Hier bekommt der Gott des Meeres lediglich Atemnot, da er nicht schwimmen kann. Infolgedessen kann sich Poseidon erstmal nur nach Westen dem Paradies zuwenden.
Besagtes Paradies ist nicht uninteressant. Da wäre zunächst einmal ein Schaf, das da dumm-dusselig durch die Prairie wandert und sich prima dazu eignet, auf einem Altar geopfert (sacrifice) zu werden. Allerdings erst, nachdem Poseidon es erlegt hat. Dazu klettert der Ex-Gott auf den Baum, wartet, bis sich das Vieh darunter zur Ruhe begibt, und hüpft dann auf den nichtsahnenden zukünftigen Lammbraten hinunter. Das Opfer wird von den drei Göttinnen, denen der Altar geweiht ist, wohlwollend angenommen. Dies bedeutet gleichzeitig Hilfe in kargen Zeiten.
Poseidon aber wendet sich nun der Pflanzenwelt zu. Mit dem Rohr könnte er zwar den Sumpf durchqueren und damit die Hydra umgehen – es kommt ihm jedoch ein besserer, geradezu göttlicher Einfall. Er legt die Blätter des Fingerhuts auf den flachen Stein und bearbeitet (pound) sie so lange mit dem kleinen Stein, bis eine giftige Flüssigkeit freigesetzt wird. Mit dieser Flüssigkeit reibt er seinen Dreizack ein, den er anschließend der Hydra zwischen die Schuppen wirft. Die wiederum schmeißt das Handtuch, und Possi kann sich als Trophäe einen ihrer hübschen Köpfe "abschnippeln".
Nachdem er durch die Pforten des Hades gewandert ist, findet sich Poseidon am Südufer des Flusses Styx wieder. Hier versucht er, einer auf die Fähre wartenden Seele den Preis für die Überfahrt abzuschwatzen. Freiwillig gibt der Typ aber nichts raus. Durch die Hilfe einer nun satten und damit wohlgelaunten Göttin jedoch kommt Possi zu mehr Geld, als er eigentlich wollte. Dies kommt ihm allerdings sehr zustatten, nachdem er ein Gespräch zwischen Charon und dem Tod belauscht und die beiden zu einer Pokerpartie herausgefordert hat.

Zocken, was des Zeusen Zeug hält

In Charons Hütte hängt Poseidon seinen auf Hochglanz polierten Schild an den Nagel in der Südwand und nimmt auf dem gegenüberstehenden Holzstuhl Platz. Auf diese fiese Art und Weise kann er Charon prima in die Karten sehen und seine eigenen Chancen etwas erhöhen. Es empfiehlt sich, den Spielstand abzuspeichern, bevor man die Hütte betritt, da man während des Spiels dazu keine Gelegenheit mehr hat.
Nun beginnt eine wilde Pokerpartie, bei der im Zweifelsfall leider immer der gewinnt, der die Regeln macht – Charon. Der wettet seine Obols ständig im gleichen Stil:

2 - 20
3 - 15
4 15
5 10
6 10
7 15
8 15
9 - 15
10 - 25
B 30
D 30
K - 30
As - 35

Diese Methode kann man ohne weiteres übernehmen. Das Spiel zieht sich dann aber endlos hin, und der Ausgang ist ungewiß. Schneller geht es, wenn man hoch setzt (30, 40 oder 50 Obols), damit Charon schnell und gründlich pleite geht und zur Tilgung seiner Spielschulden eine Einfuhrgenehmigung für die Schlüssel im Wandschrank ausschreibt. Falls Charon nach dem ersten Spiel noch nicht pleite genug ist, kann ihn Poseidon nach der nächsten Überfahrt einfach nochmal herausfordern, und dann sollte es eigentlich klappen.
Die Schlüssel im Schrank sind nur nach der auf dem Zettel notierten Methode auf die andere Seite des Flusses zu bringen. Also:
1. Den großen und den kleinen goldenen Schlüssel nehmen. Zuerst den kleinen auf dem Nordufer, dann den großen auf dem Schiff ablegen.
2. Nun die übrigen Schlüssel einzeln und in der angegebenen Reihenfolge hinüberbringen:
- Silber groß -
- Silber klein -
- Bronze groß -
- Bronze klein
3. Nun noch den großen goldenen Schlüssel vom Schiff mitnehmen, und dem Eintritt in das Schloß steht fast nichts mehr entgegen.

Endspurt im Schloß

Dieses "fast" entpuppt sich als heruntergelassenes Fallgitter, das den Zugang zu Hades' Residenz verwehrt. Es ist sehr leicht zu öffnen, wenn man an dem Seil zieht, das westlich vom Eingang an der Mauer baumelt, nur leider ist dieses Seil nicht lang genug, um bis zum Eingang selbst zu reichen. Possi ist für einen Gott aber gar nicht dumm. Ihm sind die am Himmel kreisenden Aasgeier aufgefallen, und nachdem eine Ration Futter schon einmal geholfen hat, könnte der Trick ja ein zweites Mal funktionieren. Also bindet er den Hydrakopf an das Seil, läßt sowohl Seil als auch Kopf los und begibt sich zum Tor. Einer der Geier stürzt sich auch sofort auf den Kadaver, fliegt damit weg und öffnet somit für kurze Zeit das Tor, bevor das Seil reißt. Die Zeitspanne ist aber lang genug, um auch einen etwas kurzatmigen Gott unter dem Fallgitter durchflitzen zu lassen.

Nun muß Possi nur noch die diversen Türen mit den entsprechenden Schlüsseln aufschließen. Leider nimmt das Licht immer mehr ab, so daß er sich gezwungen sieht, zuerst seinen Schild in die physikalisch gut plazierte Rinne des ersten Steinblocks zu stellen und dann den wunderschönen Spiegel in zwei Teile zu zerdeppern, die er auf den anderen beiden Quadern unterbringt. So wird genug Licht in den letzten, absolut düsteren Raum geleitet, wo Possi sich den hart erkämpften Hut nur noch zu nehmen braucht.

INFIDEL – die Hieroglyphen

Bei dem Infocom-Adventure Infidel ist eine Lösung beinahe unmöglich, wenn man die Inschriften nicht entziffern kann, die es in vielen Räumen am Boden oder an der Wand zu lesen gibt. Mit dieser fast kompletten Übersetzung soll nun hoffnungslos festgefahrenen Abenteurern geholfen werden.

...>	gehe
<...	kehre zurück
*->	nimm, nimm weg
<-*	lege, stelle, benutze
>>	schau, drehe
(	öffne
-!-	bringe ins Gleichgewicht
!	verborgen
.	unter
. -	über
-.	östlich, Osten
.-	westlich, Westen
\ .	südwestliche Ecke
/ .	südöstliche Ecke
. \	nordöstliche Ecke
. /	nordwestliche Ecke
(.)	zwischen
>...	bei
/	zu, nach
<.>	erster
<:.>	dritter
<::.>	fünfter
(@)	Nr.1
(@@)	Nr.2
(@@@)	Nr.3
(@@@@)	Nr.4
#	der, die, das
::	und
...	dann
<<>>	alle
!@!	Königin, königlich
//\ \	Schätze
/!\	Mast, Pfosten
=!=	Haufen, Büschel
= =	Buch
!*	goldener Kelch
*!	silberner Kelch
>*>	Scarabäus
!!!	Hindernis
(= =)	Fußboden
! ! !---!	große Öffnung
! ! !--!	kleine Öffnung
!=! ! !	Türsturz, Türöffnung
- ! ! -	Tür
!-! ! ! !-!	Loch, Abgrund

A: Throne Room
B: Anteroom
C: Small Anteroom
D: Brightly decorated Corridor
E: Dank Passageway
F: Grand Hallway
G: Long sloping Corridor
H: Outside Hades' Palace
I: Outside Hades' Palace
J: Punishment Fields
K: Plain of Asphodel
L: Elysian Fields
M: Plateau
N: Beside River Styx
O: Wooden Jetty
P: On the Ferry
Q: Wooden Jetty
R: Wooden Shack
S: Beside River Styx
T: Swamp
U: Beneath Swamp
V: Hades' Gates
W: Up the Tree
X: Tranquil Garden
Y: Outside Hades' Gates

Ricochet

Von Holger Wendenburg erhielten wir wichtige Hinweise, die sich auf die Kassettenversion von Ricochet beziehen.

Drückt man während des Spieles die TAB-Taste, so gelangt man jeweils einen Screen weiter, ohne daß Punkte oder Leben verloren werden.

Bei Benutzung der DEL-Taste bleibt man im aktuellen Screen, doch geht die Aufprallzahl auf Null zurück. Es kann allerdings vorkommen, daß bei zu häufiger Benutzung dieser Taste das Spiel aussteigt.

Trantor the last Stormtrooper

Cheat-Modus und Passwörter zu diesem Spiel hat Bernhard Seitz herausgefunden.

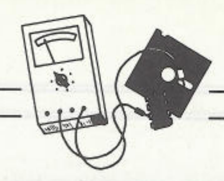
Der Cheat-Modus:

An einem vollen Schrank muß der Joystick nach unten gedrückt werden. Verharren Sie in dieser Position, und drücken Sie gleichzeitig die Tasten P R O B E.

Die Passwörter:

Cailrins - Sinclair
Beadoryk - Keyboard
Efwsorta - Software
Etopsnmk - Kempston
Mcrpuest - Spectrum
Lamentir - Terminal
Statsece - Cassette
Roadswsp - Password
Kostjyci - Joystick
Pmtcoure - Computer
Redrawha - Hardware
Carhipsg - Graphics

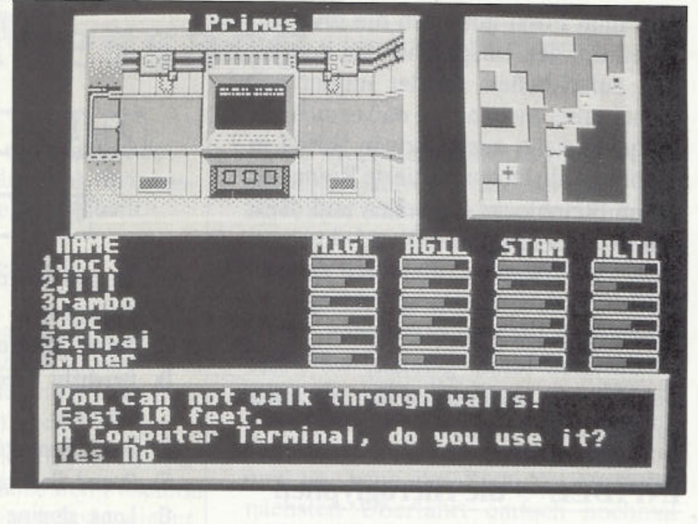
(Antje Hink/jf)



Mines of Titan

Proscenium meldet sich nicht

Im 22. Jahrhundert ist es der Menschheit gelungen, Saturns Mond Titan wohnlich zu gestalten. In unterirdisch angelegten Städten findet man alles, was das Herz einer Person, die es zum Saturn zieht, begehrt. In diese mehr oder weniger idyllische Atmosphäre dringt nun die Nachricht, daß jeglicher Kontakt zur Stadt Proscenium abgebrochen ist.



Da die Gerüchteküche bereits zu brodeln beginnt, wird es unabdinglich, herauszufinden, was genau zu dieser nicht geplanten Unterbrechung aller Funkverbindungen geführt hat.
So machen Sie sich auf, eine Gruppe von Mitstreitern zu sammeln, um gemeinsam die Fähigkeiten zu erlangen, diese schwierige Aufgabe zu erfüllen.
Nach dem letztlich doch enttäuschenden Battletech ist Mines of Titan der zweite Streich der Westwood-Infocom-Zusammenarbeit. Wieder handelt es sich dabei um ein Rollenspiel, das natürlich im Vergleich zum Erstlingswerk mit einigen Neuerungen und Erweiterungen aufwartet. Wie üblich hat der Spieler eine Party zusammenzustellen, mit der er Kämpfe und Missionen erfolgreich zu bestehen hat, bis am Ende das gesetzte Ziel, die Aufklärung der Frage nach der Unterbrechung des Kontaktes zur Stadt Proscenium, puzzlestückhaft zusammengesetzt wird.
Wie schon der Vorgänger wird auch Mines of Titan komplett mittels Menüsystem gesteuert. Alle Aktionen, die möglich sind, werden in entsprechenden Listen angewählt, was den großen Nachteil hat, daß besondere Aktionen alleine durch ihr Auftauchen in den Listen erkannt und nicht etwa gedanklich vorbereitet werden müssen.
Grafisch bietet das Programm alles, was man sich als Rollenspieler wünscht: Blick auf das Szenario, Auto-Mapping, Einblick in alle bisher begangenen Gebiete etc.
In spielerischer Hinsicht dagegen kann man geteilter Meinung sein. Gibt es auf der einen Seite unzählige Orte, die man besuchen und erforschen kann, so fällt die Art und Weise, wie man tätliche Auseinandersetzungen bestreitet, eher negativ auf. Hier hat der Spieler stets zwei Möglichkeiten. Er kann den Computer den Kampf selbständig durchführen lassen, was sich aufgrund des recht harakirischen Kampfstils nur in Kämpfen mit stark unterlegenen Gegnern empfiehlt. So bleibt nur die taktische Auseinandersetzung, bei der der Spieler jedes Party-Mitglied einzeln instruiert. Für eine ausgefeilte Strategie liegen dann aber zu wenig einblickbare Informationen zugrunde, so daß man nach wenigen "Schlägereien" den Spaß daran verlieren kann.

Infocom, bleib bei Deinen Textadventures

Wie schon bei Battletech müssen alle Freunde von "normalen" Infocom-Textadventures auch vor Mines of Titan gewarnt werden.
Das Programm zeigt sich im Vergleich zum Erstlingswerk zwar in fast allen Punkten verbessert, wirkt aber letztlich wie ein mittelmäßiger Abklatsch bekannter Rollenspiele wie Wasteland, Ultima oder den Rollenspielschlachtschiffen von SSI.
Wer Freude an detailbesessenen Abenteuerspielen hat, der wird mit Mines of Titan sicherlich nicht das große Glückslos ziehen.
Trotzdem soll das Programm an dieser Stelle nicht ganz schlecht gemacht werden.

(jf)

Hersteller: Westwood/Infocom
Preis: ca. 89 DM
Besonderheiten: Nicht kopiergeschützt, leicht auf Festplatte installierbar. Unterstützt außer Hercules alle Grafikmodi, vor Raubkopierern durch Codewortabfrage gesichert.



Ein Helferlein fürs Batchen

Viele werden das Problem kennen. Man hat sich ein wunderschönes Menü-Programm mit Hilfe der eingebauten Batchbefehle geschrieben. Dann soll ein bestimmtes Programm von Diskette gestartet werden. Doch, wie der Zufall so spielt, das Laufwerk ist nicht richtig verriegelt, oder man vergißt die Diskette einzulegen. Was macht das DOS? Mitten in dem schönen Menü prangt nun die lapidare Meldung: Laufwerk nicht bereit. (A)bbruch, (W)iederholen, (U)ebergehen.

```
Auswahlmenue
(1)  Inhaltsverzeichnis zeigen
(2)  Verzeichnis wechseln
not ready reading drive B
Abort, Retry, Fail?ectory-Tree zeigen
(4)  Dateien laden
(5)  Multi-Screen-Anzeige
(6)  Untermenue 1:  Funktionen
(7)  Untermenue 2:  Editor
```

Die Zerstörung eines einstmals guten Bildschirmaufbaus durch eine DOS-Fehlermeldung. Wir schaffen Abhilfe

Doch halt, wer nun verzweifelt abwinkt und meint: "So ist halt das Leben", dem naht Hilfe. Das hier vorgestellte Programm prüft mit Hilfe von BIOS-Funktionen, ob sich eine Diskette im Laufwerk A oder B befindet. Optional besteht die Möglichkeit zu prüfen, ob die eingelegte Diskette ein bestimmtes Label hat. Einsatzgebiet dieses in C geschriebenen Programms sind hauptsächlich Batch-Dateien, in denen es darauf ankommt, daß sich bestimmte Disketten in bestimmten Laufwerken befinden. Das bekannte 'IF EXIST' von DOS läßt sich meist nicht verwenden, wenn nicht absolut sichergestellt ist, daß sich auch eine Diskette im Laufwerk befindet.

Wenn's DOS nicht kann, dann halt übers BIOS

Um die bekannte Fehlermeldung von MS-DOS (bei einem nicht bereiten Laufwerk, beschädigter Diskette oder ähnlichem Ungemach) zu vermeiden, wird bei getdisk von BIOS-Funktionen Gebrauch gemacht. Bei TURBO C geschieht dies der Einfachheit halber über die Bibliotheksfunktion biosdisk (Parameter und sonstiges können Sie Ihrem Compilerhandbuch entnehmen). Das Programm läuft folgendermaßen ab:

1. Überprüfung auf richtige Parameter

2. Überprüfung, ob sich eine Diskette im angegebenen Laufwerk befindet. Dies geschieht mit Hilfe der BIOS-Funktion VERIFY. Es wird überprüft, ob sich der erste Sektor der Diskette lesen läßt.

3. Ist eine Diskette im Laufwerk und wurde ein bestimmter Name als zweiter Parameter angegeben, wird mit Hilfe der Bibliotheksfunktion findfirst versucht festzustellen, ob die Diskette diesen Namen hat.

Dieser ganze Vorgang wird aber erst gestartet, wenn der Bediener eine Taste drückt (man kann dann vorher eine Aufforderung zum Einlegen der Diskette auf den Bildschirm bringen). Betätigt man die Taste 'Q', wird das Programm sofort abgebrochen. Innerhalb einer Batchdatei läßt sich mit Hilfe der Anweisung 'IF ERRORLEVEL==' genau feststellen, was das Programm gemacht hat.

Die Errorlevel:

0 Kein Fehler
1 Diskette hat falschen Namen
2 Keine Disk im Laufwerk
3 Diskname zu lang (>11 Buchstaben)
4 Falsches Laufwerk
5 Keine Parameter
6 Programm vom Benutzer abgebrochen

Ein Tip: DOS vergleicht bei der Anweisung
IF ERRORLEVEL==2 ....
ob Errorlevel gleich oder größer 2 ist, nicht, ob es genau 2 ist. Also muß mit dem größten möglichen ERRORLEVEL begonnen werden, etwa:
IF ERRORLEVEL==6 ...
IF ERRORLEVEL==5 ...
....

Einschränkungen und sonstiges

Das Programm erkennt nur A und B als Laufwerke an. Das liegt hauptsächlich daran, daß die Laufwerke ab C ohnehin meist Festplatten oder RAM-Disks sind, und die lassen sich auch mit IF EXIST überprüfen. Bei den möglichen Laufwerken A und B handelt es sich um physikalische Laufwerke.

Ein anderes Problem stellt der Diskettenname dar. Manchen mag beim Überfliegen des Listings ein Programmteil aufgefallen sein. Dort wird der Diskettenname mit den Laufwerksbuchstaben zusammengebaut und, falls der Name größer als acht Buchstaben ist, ein Punkt an der neunten Stelle eingefügt. Obgleich der Name, bei der Namensgebung mit Format oder Label, nicht mit einem Punkt versehen wird, baut das DOS bei der Funktion findfirst diesen Punkt an der neunten Stelle ein, wenn der Name länger als acht Buchstaben ist.

Zum Abschluß sei noch auf eine nette Eigenschaft von TURBO C hingewiesen. Hat der Diskettenname ein Leerzeichen, geben Sie beispielsweise ein: getdisk a: "Disk 1"
Die Anführungszeichen veranlassen TURBO C, den dazwischenliegenden String als einen Parameter anzusehen. So kann man sich eine Menge Arbeit beim Zusammensetzen von eigentlich zusammengehörigen Teilen eines Strings aus der Argumentzeile sparen.

(Robert Haas/jf)

Listing-Info:

Das Listing wurde in der Programmiersprache Turbo C entwickelt.
Tippen Sie das Programm mit einem normalen Texteditor (etwa ConText, Word etc.) ab. Sichern Sie den Code als ASCII-Datei und compilieren Sie das ganze mit dem Turbo C-Compiler.

Auf der Databox befindet sich eine bereits compilierte Fassung dieses Programms.




```
/*  Dieses Programm stellt fest, ob sich in einem angegebenen Laufwerk
eine Diskette befindet.  Optional kann  auch  der  Name  der  Diskette
angegeben werden

Compiler    :  Turbo C 2.0 (Borland)
Entwickelt :  28.08.1989 von  Robert Haas, 1989
Letzte Änderung : 09.10.1989

Aufruf : getdisk drivename[:] [diskname]

Argumente in eckigen Klammern sind optional.
*/

#include <bios.h>                    /* Headerfiles mit Prototypen */
#include <conio.h>
#include <stdio.h>
#include <dos.h>
#include <string.h>
#include <process.h>
#include <ctype.h>
#include <dir.h>


char  usage[MAXPATH+10],             /* Name des Programms (argv[0]) */
      diskname[20],                  /* Zum Zusammensetzen des Disk- */
                                     /* namens                       */
      biosbuffer[512],               /* Buffer für biosdisk          */
      copyrigth[] = "Robert Haas, 1989";

                                     /* Prototyp der Fehlerausgabe-  */
                                     /* funktion                     */
void error( char *meldung1,char *meldung2, int errorcode );


int main( register argc, register char **argv)
{
    int          drive, try;         /* Laufwerknummer, Versuchsanzahl*/
    struct ffblk fb;                 /* struct für findfirst         */

    strcpy( usage, "\nUsage: " ); /* Name des Programms zusammen- */
    strcat( usage, argv[0] );     /* bauen                        */

    if( argc==1 )                  /* Mindestens ein Argument      */
        error( "\nError! No Parameter(s).", "" , 5);

    if( strlen( argv[1] ) > 2 )    /* Gültiger Laufwerksname?      */
        error( "\nError! Bad Drivename.", argv[1], 4 );
    if( ( strlen( argv[1] ) > 1 ) && ( argv[1][1] != ':' ) )
        error( "\nError! Bad Drivename.", argv[1], 4 );
    if((toupper(argv[1][0]) != 'A') && (toupper(argv[1][0]) != 'B'))
     error( "\nError! Bad Drivename. Only Drives A and B.",argv[1],4);

                                   /* Der Diskname muß <= 11 Buch- */
                                   /* staben sein.                 */
    if(( argc==3 ) && ( strlen( argv[2] ) > 11 ))
        error( "\nError! Bad Diskname.", argv[2], 3 );

                                   /* Buchstabe in Laufwerksnummer */
```

Listing HELFER

```
    drive = toupper( argv[1][0] ) - 65;

                                   /* Auf Tastendruck warten.Q==Ende*/
    if( toupper( getch() ) == 'Q')
        return 6;

                                   /* Drei versuche Disk zu lesen   */
    try = 3;
    do                             /* Sektor - Verify, ( BIOS )     */
    {
      if( biosdisk( 4, drive, 0, 0, 1, 1, biosbuffer ) != 0 )
      {
      /* Wenn Fehler beim Lesen, Disksystem zurücksetzten ( BIOS ) */
        biosdisk( 0, drive, 0, 0, 1, 1, biosbuffer );
      }
      else                         /* Kein Fehler, weitergeht's     */
        break;
    }while(--try);
    if( try == 0 )                 /* Nach 3 Versuchen, Abbruch mit */
        return 2;                  /* Errorlevel 2                  */
    if( argc == 3 )                /* Bei 3 Argumenten ist das 3.   */
    {                              /* der Disketten Name            */
        strcpy( diskname, argv[1] ); /*Zusammenbasteln von Lauf-    */
        if( strlen( diskname ) < 2 ) /*werk und Disketten Namen     */
            strcat( diskname, ":" );
        strcat( diskname, "\\*.*" );
                                    /* DOS-Funktion: ersten,        */
                                    /* passenden Dir-Eintrag finden */
        if( findfirst( diskname, &fb, FA_LABEL ) == -1 )
            return 1;               /* Kein Name                    */
        else
        if( strlen( argv[2] ) > 8 )  /* siehe Text                  */
        {
            register i = 0;          /* Zählervariable              */
            try = 0;
            while( argv[2][i] )
            {
                diskname[try++] = argv[2][i++];
                if( i==8 )
                    diskname[try++]='.';
            }
            diskname[try]='\0';
            if( stricmp( diskname, fb.ff_name ) )
                return 1;            /* Falscher Name               */
        }
    }
    return 0;                        /* Alles OK                    */
}

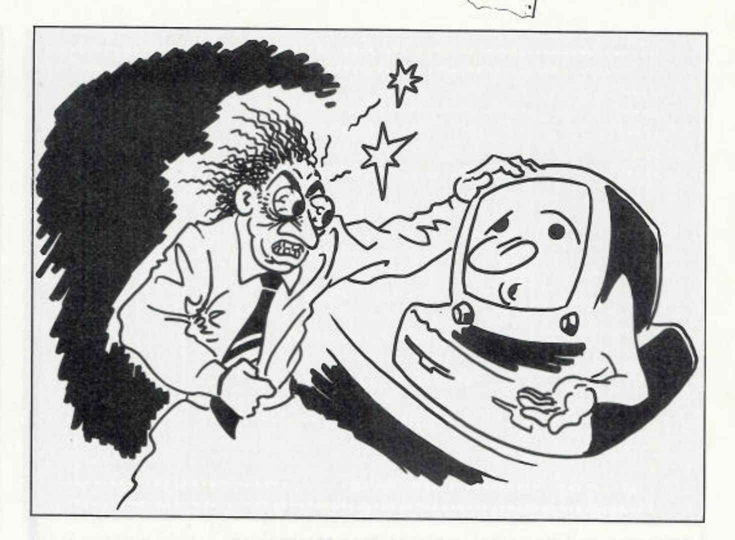
/* Hier werden alle Fehlermeldungen ausgegeben und das Programm
mit dem entsprechenden Errorlevel-Code beendet */

void error( char *m1, char *m2, int code )
{
    fprintf( stderr, "\n%s %s%s drive[:] [diskname]"
    "\nParameters in [] are optional\n", m1, m2, usage );
    exit( code );
}
```

Listing HELFER

Patzige Datensicherung

Datenschutz auf dem Vormarsch



In einer Zeit, in der immer mehr Arbeiten und Dienstleistungen vom Computer verrichtet werden, häufen sich zwangsläufig auch die Datenmengen, deren Wichtigkeit den Benutzer zwingt, sie vor der Einsicht anderer Personen zu schützen. Wir liefern Ihnen eine softwaremäßige Lösung, die für Dritte nicht gleich erkennbar ist und darüber hinaus durch einige Nebeneffekte zugangsunberechtigte Personen zur Weisglut bringen kann.

Das Programm Security simuliert eine äußerst patzige MS-DOS-Oberfläche, was heißen soll, daß sich Security nach dem Start zwar mit einem gängigen Prompt (C:>) meldet und zusätzlich die wichtigsten DOS-Befehle erkennt, diese aber auf eine sehr eigene Weise interpretiert und kommentiert. Durch Eingabe eines Passwortes gelangt der rechtmäßige Benutzer auf die normale Betriebssystemebene zurück.

Um eine nahezu hundertprozentige Sicherheit zu erreichen, wird Security als erster Befehl in die AUTOEXEC.BAT eingetragen. So wird gewährleistet, daß ein eventuelles Unterbrechen des Bootvorgangs durch Drücken von CTRL-C fast unmöglich wird. Ist das Programm erst einmal im Speicher vorhanden, so läuft nichts mehr, falls man nicht das Passwort kennt. Die Hauptaufgabe des Programms besteht dann auch darin, Unterbrechungen des Benutzers zu ignorieren. Auswirkungen, die das Drücken der CTRL-C-Taste auf das laufende Programm haben, werden mit dem Checkbreak-Befehl unter Turbo Pascal unterbunden.

Das weitaus größere Problem der Unterbindung von Konsequenzen, die das Drücken von CTRL-Break verursacht, wird durch eine kleine Interruptverbiegung gelöst. Der hierfür verantwortliche Interrupt 1B Hex wird auf die Prozedur Break__weg gelegt und unschädlich gemacht.

Die Nebenwirkungen

Neben der Hauptaufgabe des Schutzes der eigenen Datenbestände haben wir dem Programm einige Extras mitgegeben. Das Programm schaut sich Ihre Eingaben ein und vergleicht diese mit einer erweiterbaren Liste von Eingabemöglichkeiten. Sollte Ihr Befehl in der Liste vorhanden sein, so wird eine Reaktion ausgelöst. Meist handelt es sich dabei um eine patzige verbale Äußerung des Rechners. Sie können die Antworten des Computers natürlich nach Belieben verändern. Dazu müssen Sie die Werte der Konstanten am Programmanfang anpassen. Beachten Sie bitte ebenfalls, daß Erweiterungen oder Änderungen in der Liste der Begriffe, die das Programm erkennt, in Großbuchstaben zu erfolgen hat. Dies ist deshalb wichtig, weil alle von Ihnen getätigten Eingaben in Großbuchstaben verwandelt werden.

Security erkennt auch, ob eine Funktionstaste gedrückt wurde, so daß auch hier Konsequenzen programmiert werden können.

Das Listing enthält eine abgespeckte Version dieses Programms. Sie können so die Reaktionen des Programms ganz nach Ihren Vorstellungen beeinflussen. Auf der Databox befindet sich neben dieser kleinen Version eine ausgereifte, die Sie gleich benutzen können.

Sollten Sie keinen Turbo-Pascal-Compiler 5.0 besitzen, so sind Sie trotzdem in der Lage, das Passwort zu verändern. Mittels File-Editor können Sie das fertige Programm (SCHUTZ.EXE) der Databox geringfügig verändern. Fast jede Benutzeroberfläche besitzt eine solche View/Edit-Funktion, mit der in beliebigen Dateien herumgestöbert werden kann. Suchen Sie in SCHUTZ.EXE den Ausdruck "0123456789". Dieser steht stellvertretend für das Passwort. Schreiben Sie in diese zehn Stellen Ihre persönliche Eintrittsberechtigung, wobei Sie nur Ziffern und Großbuchstaben verwenden dürfen. Und denken Sie bitte daran, nicht den Namen Ihrer Freundin oder den des getreuen Dackels zu verwenden, wie es zerstreute Professoren gerne zu tun pflegen. Sichere Passworte sind solche, deren Zusammenhang nicht zu erschließen ist, etwa eine zufällige zehnstellige Zahlenkolonne. Geburtstage, Lottozahlen oder sinnvolle Worte sind zwar einfach zu merken, aber meist ebenso einfach herauszubekommen.

Vor- und Nachteile der totalen Sicherheit

Wenn Sie dieses Programm auf jeder Ihnen zur Verfügung stehenden Systemdiskette installieren, so erhalten Sie einen nahezu perfekten Schutz. Als Nachteil erweist sich hier, daß Sie größte Schwierigkeiten bekommen werden, sollten Sie einmal das Passwort vergessen. Dann läuft nichts mehr! Heben Sie daher in einer der hinteren Ecken Ihres Diskettenbestandes in jedem Fall eine normale Systemdiskette auf. So haben Sie die Sicherheit, daß Sie auch in einem solch fatalen Unglücksfall auf Ihren Computer zurückgreifen können.

(jf)

Listing-Info:

Die 'Patzige Datensicherung' wurde inTurbo-Pascal 5.0 geschrieben.

Zum Abtippen und Compilieren benutzen Sie bitte diesen Turbo-Pascal-Compiler.




```
PROGRAM SCHUTZ;

USES TURBO3,CRT,DOS;

CONST ende      =  8;
      zufallmax =  3;
      breckmax  =  2;

TYPE Wort = STRING[70];

VAR  a            : Wort;
     ok           : BOOLEAN;
     feld         : ARRAY[1..ende] OF Wort;
     meldung      : ARRAY[1..zufallmax] OF Wort;
     breck        : ARRAY[1..breckmax] OF Wort;
     ein          : CHAR;
     i            : INTEGER;
     notausgang   : wort;

{------------------------------------------------------------------------}
PROCEDURE Defis;                    { Das Array feld enthält alle Begriffe, }
  BEGIN                             { die das Programm erkennt. Sie können  }
  feld[1]:='DIR';                   { hier beliebige Ergänzungen eintragen, }
  feld[2]:='PATH';                  { sofern Sie diese in Großbuchstaben    }
  feld[3]:='CLS';                   { schreiben und die Konstante ende auf  }
  feld[4]:='CHKDSK';                { den neuen Wert einstellen.            }
  feld[5]:='DELETE';                { Gleiches gilt für die Arrays meldung  }
  feld[6]:='FIND';                  { und breck (meldung enthält alle Ant-  }
  feld[7]:='LABEL';                 { worten, die bei nicht bekannten Ein-  }
  feld[8]:='TYPE';                  { gaben gegeben werden, in breck stehen }
                                    { diejenigen, die ausgegeben werden,    }
                                    { wenn CTRL-Break gedrückt wurde.       }

  meldung[1]:='Kenn ich nich, will ich nich, mach ich nich!';
  meldung[2]:='Hatte da jemand was gesacht?';
  meldung[3]:='Der Befehl geht mir ja derart am seriellen Port vorbei';
  breck[1]:='Ui Ui, was war denn das?';
  breck[2]:='Tztztztztztztztztztztztztztztzt';
  END;
{------------------------------------------------------------------------}


{------------------------------------------------------------------------}
PROCEDURE Retour(VAR i:INTEGER);
  VAR a,x,y : INTEGER;
  BEGIN
  CASE i OF
    1 : WRITELN('Wie ich DIR, so du MICH');
    2 : WRITELN('Viele Wege führen nach Rom, das hier ist der Falsche');
    3 : BEGIN
        WRITELN('FATAL ERROR ON SCREENOVERDUP SECTION B345');
        FOR a:=1 TO 500 DO
          BEGIN
          x:=RANDOM(80);
          y:=RANDOM(25);
          GOTOXY(x,y);
          WRITE(CHR(RANDOM(50)+52));
          END;
        END;
    4 : WRITELN('Untersuche Kreis ???');
    5 : WRITELN('DELETE??? Das heißt Das Lied, Mänsch!');
    6 : WRITELN('Aber ich bin doch Dein Freund!');
    7 : WRITELN('Marlow, ich fleh Sie an Marlow, finden Sie LABEL...');
    8 : WRITELN('Nein, Du bist absolut nicht mein TYPE');
  END; {end of case}
  END;
{------------------------------------------------------------------------}


{------------------------------------------------------------------------}
PROCEDURE Break_weg;INTERRUPT;         { Auf diese Prozedur wird der In- }
  VAR   zu : INTEGER;                  { terrupt gelegt, der normaler-   }
  BEGIN                                { weise durch CTRL-Break ausge-   }
  zu:=RANDOM(breckmax)+1;              { löst wird. Der Befehl INTERRUPT }
  WRITELN(breck[zu]);                  { muß benutzt werden, damit eine  }
  END;                                 { korrekte Rückkehr (IRET) ge-    }
                                       { währleistet ist.                }
{------------------------------------------------------------------------}


{------------------------------------------------------------------------}
PROCEDURE Gross;                       { Alle Eingaben werden in Groß-   }
  VAR  i,j : INTEGER;                  { buchstaben verwandelt. Dies     }
       ch  : CHAR;                     { erleichtert die Abfragen erheb- }
  BEGIN                                { lich.                           }
  j:=LENGTH(a);
  FOR i:=1 TO j DO
    BEGIN
    ch:=a[i];
    a[i]:=UPCASE(ch);
    END;
  END;
{------------------------------------------------------------------------}


{------------------------------------------------------------------------}
PROCEDURE Kont;
  VAR i    : INTEGER;
      ko   : BOOLEAN;
  BEGIN
  ko:=FALSE;
  i:=0;
  REPEAT                               { Die Liste wird durchgearbeitet  }
    i:=i+1;                            { bis ein bekannter Ausdruck er-  }
    IF a=feld[i] THEN                  { reicht wird.                    }
      BEGIN
      Retour(i);
      ko:=TRUE
      END;
  UNTIL ko OR (i>=ende);
  IF NOT ko THEN WRITELN(meldung[random(zufallmax)+1]);
  IF a=notausgang THEN ok:=TRUE;    { Ist die Eingabe das Passwort?   }
  END;
{------------------------------------------------------------------------}


{------------------------------------------------------------------------}
PROCEDURE Fun (i:INTEGER);            { Antworten, wenn eine Funktions- }
  BEGIN                               { taste gedrückt wurde.           }
  CASE i OF
   59 : WRITELN('Sollte ich das erste Zeichen vergessen haben?');
   60 : WRITELN('Diese Funktion habe ich sowieso noch nie begriffen');
   61 : WRITELN('Auch noch faul, was? Nochmal eintippen, klar?');
   62 : WRITELN('Finga fon die Eff-fier-Taste');
   64 : CLRSCR;
   END; {case}
   END;
{------------------------------------------------------------------------}


{------------------------------------------------------------------------}
{                          HAUPTPROGRAMM                                 }
{------------------------------------------------------------------------}
BEGIN
notausgang:='0123456789';             { Hier steht das Passwort          }
SETINTVEC($1b,@break_weg);
CHECKBREAK:=FALSE;
CLRSCR;
Defis;
ok:=FALSE;
REPEAT
a:='';
WRITE('C:>');
REPEAT
  READ(KBD,ein);
  IF (ORD(ein)>0) AND NOT (ORD(ein)=13) THEN
    IF ORD(ein)=27 THEN
      BEGIN                           { Eine Funktionstaste wurde ge-    }
      ein:=READKEY;                   { drückt. Welche?                  }
      Fun(ORD(ein));
      END
    ELSE
      BEGIN
      a:=a+ein;
      WRITE(ein);
      END;
UNTIL ORD(ein)=13;                    { Return wurde gedrückt            }
WRITELN;
Gross;
Kont;
UNTIL ok;                             { ok wird true, wenn das Passwort  }
END.                                  { richtig eingegeben wurde.        }
{------------------------------------------------------------------------}
{------------------------------------------------------------------------}
```

Listing SCHUTZ

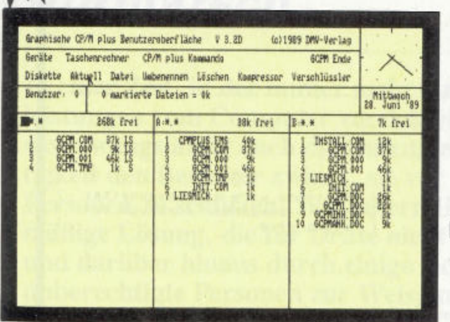
Joyce Programmsammlungen

***Hochwertige Software** zu **Niedrigpreisen** finden JOYCE-Besitzer im Rahmen einer Programmsammlung in der Angebotspalette des DMV Verlages.*

Jede Ausgabe aus dieser Reihe enthält eines oder zwei Programme, die aus verschiedenen Anwendungsgebieten kommen. Diese Serie erscheint in unregelmäßiger Reihenfolge und wird als komplettes Programmpaket mit 3"-Diskette und Bedienungsanleitung ausgeliefert.

Jetzt neu

Joyce-Programmsammlung VOL. VII

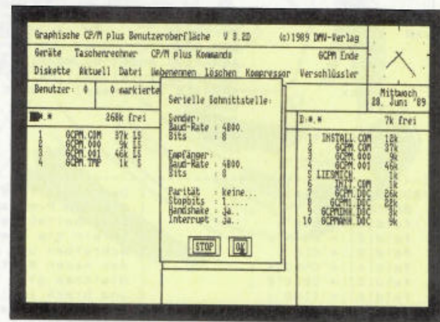


Keine Inhaltsverzeichnisse mehr mühsam einzeln auslesen! Mit GCPM sind Sie über alle Laufwerke immer im Bilde

Arbeiten Sie schnell, bequem und komfortabel mit

GCPM - der grafischen Benutzeroberfläche für den Joyce.

Beneiden Sie nicht länger die PC-Besitzer! GCPM bietet Ihnen nun auch für den Joyce eine ausgefeilte grafische Benutzeroberfläche inklusive Mausbedienung, Uhr, Taschenrechner, Schnittstellensteuerung, Passwortabfrage und vielem mehr.



So muß Computerbedienung sein! Auch die serielle Schnittstelle läßt sich in einem Fenster schnell und fehlerfrei einstellen

GCPM bietet Ihnen unter anderem:

Steuern Sie alle Funktionen Ihres Joyce wahlweise über Maus oder Tastatur.

Arbeiten Sie mit der Darstellung aller Dateien in Tabellenform ähnlich LocoScript und den Standardfunktionen wie Kopieren, Umbenennen und Löschen.

Auf bis zu drei Laufwerken können die Parameter, Suchpfade (ohne SETDEF) und das temporäre Laufwerk beliebig eingestellt werden.

Verändern Sie die Dateiattribute, und rufen Sie Programme direkt von GCPM aus auf.

Nutzen Sie die Funktionen zum Verschlüsseln und Komprimieren von ASCII-Dateien und die optionale Passwortabfrage

Sparen Sie Zeit mit dem integrierten Taschenrechner mit Zwischenspeicher und umfassenden Rechenfunktionen.

Verändern Sie beliebig die Grundeinstellungen der Schnittstelle, des Zeichensatzes, der Tastatur oder Maus, der Diskettenlaufwerke und des Druckers.

Ein Screensaver stellt den Monitor bei Nichtgebrauch dunkel.

Mit dem GCPM-Starterset als Installationsprogramm und mit 40seitigem Handbuch mit vielen Abbildungen können Sie im Handumdrehen GCPM auch auf Ihrem Joyce anwenden.

Ein Programm, das jeder Joyce-Besitzer haben muß!

Joyce-Programmsammlung VOL. VII:
GCPM - grafische Benutzeroberfläche
(JOYCE PCW 8256/8512)

3-Zoll-Diskette **69,– DM***

VOL. VI

Tabellenkalkulation

Universell einsetzbare Tabellenkalkulation zum Erstellen von Monatsbilanzen, T-Konten etc. für die JOYCE-Familie

Funktionen:

- Kurze Einarbeitungszeit durch eine übersichtliche Menüführung, die alle Fehleingaben abfängt
- Schnelles Arbeiten durch Belegung der Funktionstasten
- Einblendung der Formel des aktuellen Feldes am unteren Rand
- Besonders schnelle Berechnungsalgorithmen
- Schnelle Schreib-/Leseoperationen durch Benutzung der RAM-Disk
- Bis zu 68 Zeichen pro Formel möglich
- Eigene Befehle zur Verkürzung der Formeln
- Drei verschiedene Schriftarten beim Drucken
- Standardmäßig 2574 frei belegbare Felder
- Akzeptiert auch Exponentenschreibweise

Vol. VI für alle JOYCE PCWs
3-Zoll-Diskette **59,– DM***

VOL. V

Datenbanksystem

Maximal 27 Felder pro Datensatz, 50 Stellen pro Feld, 35.000 Sätze pro Datei, minimaler Disketten-Speicherbedarf!

- Verwendung von Standard-Direktzugriffsdateien (BASIC)
- Automatisch generierte Maske zur Datenerfassung, Änderung und Löschung
- Auf Wunsch Datentransfer aus vorhandenen in neuangelegte Dateien
- Automatische Eintragung neuer Dateien in das Disketten-Hauptmenü
- Alle Programme werden auf der RAM-Floppy gehalten
- Listenerstellung mit automatischer Spaltenformatierung und Spaltensummen
- Freie Wahl der Sortierung; Mehrfachsortierkriterien
- Ohne Lernaufwand SOFORT voll anwendbar, keine Befehlswörter nötig
- Druckmasken für die beliebig sortierte Datenauflistung am Bildschirm oder am Drucker werden automatisch generiert
- Bei mehrseitigen Auflistungen am Bildschirm Direktsprung zu jeder Seite
- Einmal gewählte Druckparameter werden gespeichert

VOL. V für Joyce/PCW 8256/8512/9512**
***** PCW 9512 auf Anfrage***
3-Zoll-Diskette **99,– DM***

*** Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.**

DMV-Verlag · Postfach 250 · 3440 Eschwege

VOL.IV

Bild-Editor

Funktion: Grafikprogramm zum Erstellen von hochauflösenden Grafiken, welche in eigene Programme eingebunden werden können, sowie deren Ausgabe auf dem JOYCE-Drucker.

Leistungsumfang: Der mitgelieferten Diskette können Sie alle zum Arbeiten mit dem Programm benötigten Dateien entnehmen. Für alle Turbo-Pascal-Programmierer wird auch der Programmtext mitgeliefert.

Funktionen: Durch das Unterscheiden zwischen Grob- und Feinbearbeitung ist es möglich, brilliante Grafiken auf das Pixel genau zu zeichnen. Lösch- sowie Invertierfunktion sind in allen Bearbeitungsmodi aufrufbar. Auch Texte können bei der Grobbearbeitung in das Bild gebracht werden. Im Dateimenü steht neben einer Lade- und Speicheroption noch eine Funktion zur Verfügung, welche zwei Bilder miteinander mischt. Neben dem Versatz des linken Randes vor dem Drucken können noch vier verschiedene Druckformate angewählt werden.

VOL. IV – für Joyce/PCW 8256/8512

49,– DM*

VOL.III

Feld-Tab

Ein BASIC-Programm zum millimetergenauen Ausfüllen von Tabellen, Vordrucken und Formularen. Feld-Tab ist menügesteuert und bietet die Anwahl der einzelnen Funktionen wie unter LocoScript gewohnt. Geben Sie Seitenlänge, Zeilenabstand und Tabulatoren in Millimetern ein, bestimmen Sie Schriftweite, Schriftart und Text. Text kann mit LocoScript erstellt werden und nach Umwandlung in eine ASCII-Datei in Feld-Tab eingelesen werden. Weitere ASCII-Editoren können ebenso verwendet werden wie der komfortable programminterne Seiteneditor. Ein unentbehrliches Werkzeug!

Gsxplot

Ein Grafikpaket für Statistiken, Geschäftspräsentationen und viele andere grafische Anwendungen! Über ein Menü sind folgende Funktionen wählbar:
· Balkendiagramme · Kurvendiagramme · Strichgrafik
· Punktediagramme · Textgrafik
Ein Hilfsprogramm erläutert Ihnen während der Arbeit mit Gsxplot die wichtigsten Funktionen. Alle erstellten Grafiken können sowohl am Bildschirm als auch auf dem Drucker dargestellt werden. Gsxplot braucht den Vergleich mit wesentlich teurerer Software nicht zu scheuen!

VOL.III – für Joyce/PCW 8256/8512 zwei Disketten

59,– DM*

VOL.II

Super-Dateiverwaltung

Eine universelle Dateiverwaltung für PCW 8256/8512 zur Erstellung eigener Dateien.

SUPERdat ist das Hauptprogramm, welches die Daten der gewünschten Datei verwaltet. Neben der Eingabe von Daten in die Maske sind mehrere Sucharten, so z.B. auch Jokersuchen, möglich. Jede Datei kann maximal acht Felder enthalten, wovon jedes höchstens 40 Zeichen enthalten darf. Die Gesamtlänge eines Datensatzes darf 255 Zeichen betragen.

SUPERtex Dieses Programm stellt eine Rundschreib- (Mailmerge-)Funktion für SUPERdat zur Verfügung. In einen in Laufwerk M: befindlichen ASCII-Text (z.B. mit RPED erstellt) werden automatisch vom Anwender ausgewählte Einträge aus SUPERdat-Dateien an beliebiger Stelle eingefügt. Weiterhin können 30 Datensätze in eine für LocoScript lesbare Datei umgewandelt werden.

SUPERcal Der Taschenrechner zu SUPERdat. Dieser bietet neben den Grundrechenarten auch Winkelfunktionen, quadratische und Prozentfunktionen. Eine Klammerebene und Memory-Funktionen vervollständigen das Leistungsangebot dieses Programms.

VOL.II – für Joyce/PCW 8256/8512

49, – DM*

VOL.I

Der Character-Designer

Funktion: Komfortable Erstellung eigener Zeichensätze auf PCW 8256/8512 und deren Darstellung am Bildschirm! Ausdruck von ASCII-Files in diesem Zeichensatz unter CP/M Plus.
Leistungsumfang: CD.COM ist der Character-Designer, der Editierung oder Neuerstellung von Zeichensätzen und deren Speicherung erlaubt. **CD-PRINT** druckt vorformatierte ASCII-Texte auf dem Joyce-Drucker oder anderen Druckern in dem gewünschten Zeichensatz aus.
CRAZY, ORIGINAL, LOCCHAR und **SCRIPT** sind mitgelieferte Zeichensätze. **SETUP.COM** erlaubt als Zugabe die Vorwahl einiger Systemparameter, z.B. die der seriellen Schnittstelle, der Tastaturgeschwindigkeit und der Floppy-Steprate.

MGX

Funktion: Grafische Darstellung von mathematischen Funktionen und beliebigen Meßreihen auf Bildschirm oder im Großformat auf dem Drucker.
Leistungsumfang: Neben den arithmetischen Grundfunktionen sind auch weitere Funktionen darstellbar, die z.B. unter Mallard-Basic nicht zur Verfügung gestellt werden. Es können mehrere Funktionen und Meßreihen (diese wiederum mit mehreren Meßwerten gleichzeitig) dargestellt werden.

VOL.I – für Joyce/PCW 8256/8512

49,– DM*

Praktische Textverarbeitung mit Joyce



Ein Buch/Disketten-Paket. Jürgen Siebert, der Autor, zeigt in diesem Buch Möglichkeiten der Textverarbeitung auf, die Sie von Loco-Script nicht erwartet hätten... Von der Pike auf werden Sie an den Umgang mit Schablonen und Standardlayout herangeführt. Einige Abstecher führen Sie anhand anschaulicher Beispiele an Textverarbeitung und CP/M (ED/Wordstar) heran.

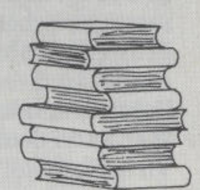
Aus dem Inhalt:
- LocoScript Spezial - Softwaretraining für Fortgeschrittene
- Fehler im System: Wie rette ich meinen Text?
- Joyce-Tasteninstallationsdatei für das Programm Wordstar
- Aleatorische Poetik: Der Computer dichtet

Auf Diskette: über 50 Dateien mit Schablonen, Briefen, Postkarten, Serien-Rundschreiben, Formularen, Etiketten, Druckbeispielen, Schriften, Bildschirm-Installationen und vieles mehr.

Leinen-Hardcover, 207 Seiten, inklusive 3-Zoll-Diskette

89,– DM*

Joyce Bücher-Kiste



Das Große LOGO-Buch zu CPC und Joyce
Data-Becker-Buch, 410 Seiten **39,– DM***

Programmierwissen pur im Westentaschenformat:

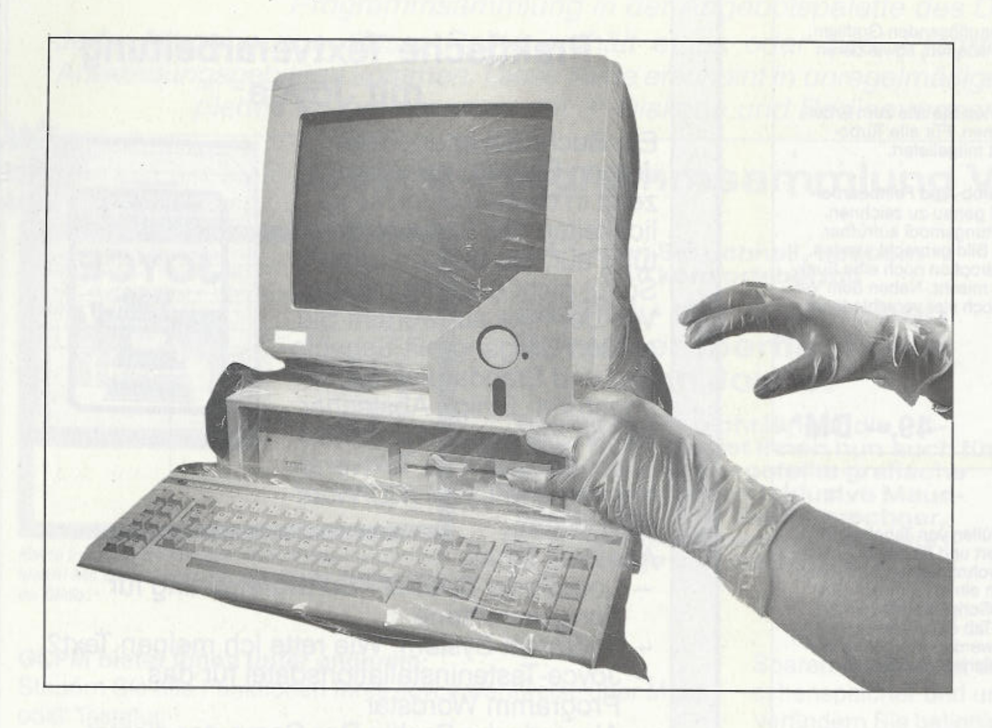
Führer zum Joyce
Data-Becker-Buch **29,80 DM***

Führer zum CP/M
Data-Becker-Buch **19,80 DM***

Den Joyce programmieren
Franzis-Verlag, ca. 160 Seiten **38,– DM*** ausverkauft

* Unabhängig von der Anzahl der bestellten Bücher berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Herr Doktor, Sagrotan

Viren, Trojaner und was man dagegen tun kann

Ein Thema, das alle Computerbesitzer in der letzten Zeit mehr und mehr beschäftigt, sind Viren. Das Problem, daß immer mehr kleingeistig destruktive Programmierer der Meinung sind, sie müßten durch die Entwicklung zerstörerischer Programme ihr Ego stärken, gehörte eigentlich in die Praxen bewährter Psychologen. Leider werden statt konzentrierter Heilung Ängste freigesetzt und gute Ideen, wie beispielsweise die der Shareware und Public Domain Software, in immer beängstigenderer Weise angegriffen. Wer kann denn noch sicher sein, daß sich auf seiner neuen PD-Disk kein Virus befindet?

Auf normalen PCs gibt es praktisch nur zwei Arten dieser Programme. Es handelt sich hierbei um 'richtige Viren' und um 'Trojanische Pferde'. Für alle, die nie den nicht nur aus Funk und Fernsehen bekannten 'Odysseus' sahen, eine kurze Erklärung zum Trojanischen Pferd.

Es ist schon ein paar Tage her...

Wie uns Homer berichtete, kam es vor langer Zeit zu einem Streit zwischen Troja und dem Rest des alten Griechenland. Es ging um eine Frau (Helena), um einen Mann (Paris) und um jemanden, der etwas gegen das Zusammensein der beiden hatte (Helenas Göttergatte Agamemnon) die übliche Dreiecksgeschichte also. Nachdem die griechischen Helden Troja ohne Erfolg einige Jahre belagert hatten, kam einer auf die schlaue Idee, ein großes hölzernes Pferd zu bauen, Soldaten darin zu verstecken, das Pferd vor die Stadttore zu stellen, und mit dem Rest der Armee so zu tun, als würde man abziehen. Man kann über die 'intellektuelle Größe' der Trojaner nur staunen, die dieses "Geschenk", ohne dessen Inhalt zu überprüfen, frisch nach dem Motto "'nem geschenkten Gaul schaut man nicht ins Maul" in ihre Stadt zogen, ein Riesenfest feierten und sich am nächsten Morgen wunderten, warum ihre Stadt zerstört war.

Zurück in die Gegenwart

Trojanische Programme werden meist über Mailboxen oder PD-Software verbreitet. (Es wird hier von einem einzelnen Computer ausgegangen, nicht von einem vernetzten System.) Ein Trojaner ist, einfach gesagt, ein Programm, das vortäuscht, eine sinnvolle Aufgabe zu haben, heimlich jedoch versucht, soviel Schaden wie irgend möglich anzurichten. Von den "Programmierern" dieser Programme wird in den Dokumentationen oder Readme-Files versprochen, es handele sich "um den besten Directory-Lister aller Zeiten" oder ähnlich Guiness-Buch-Verdächtiges. Beliebt sind auch Namen wie SEXYLADY.COM, STRIP.COM, usw. Wenn man solch ein Programm unvorsichtigerweise startet, wird es meist versuchen, auf allen erreichbaren Laufwerken die FAT zu löschen, die Diskette/Platte zu formatieren oder ähnlichen Unsinn zu veranstalten. Bei der FAT handelt es sich um den Notizblock des DOS auf der Diskette/Platte, worin vermerkt ist, welcher Sektor zu welcher Datei gehört. Wenn die FAT gelöscht ist, sind die Daten zwar noch vorhanden, ihre Zuordnung zueinander ist aber nicht mehr bekannt. Die Rekonstruktion einer Festplatte ähnelt dann dem Zusammensetzen eines Puzzles aus zirka 50000 einfarbig himmelblauen Teilen.

Erkrankungen des Computers

Während Trojanische Pferde ihre zerstörerische Wirkung recht schnell offenbaren, wenn auch der Schaden immens sein kann, sind Viren-Programme noch gefährlicher. Viren sind seit etwa 1984 bekannt. Der amerikanische Computerexperte Fred Cohen berichtete damals erstmals über Programme mit zwei typischen Eigenschaften:

1. Sie erzeugen Kopien ihrer selbst und infizieren damit andere Programme, die ihrerseits zu Virenträgern werden und bei Aktivierung weitere Programme verseuchen.
2. Diese Programme manipulieren den ursprünglichen Programmablauf.

Zum Beispiel: Zerstörung von Datenbeständen und Hardware, Ausgabe dummer Sprüche etc.

In der harten Computerrealität sieht die Infizierung eines Rechners etwa folgendermaßen aus (Ich verzichte bewußt auf zu viele Einzelheiten, um nicht schuld daran zu sein, daß sich jemand "berufen" fühlt, so etwas zu probieren und noch mehr dieser "Tierchen" in Umlauf geraten.):

Der Computerbesitzer Hubert P. aus M. hat von einem Bekannten einen Stapel Disketten mit den neuesten PD-Programmen bekommen, die dieser sich aus einer Mailbox geholt hatte. Voller Freude schaltet er seinen PC ein, schiebt die erste Diskette hinein und startet eines der Programme. Was er nicht weiß: Ausgerechnet dieses

Programm wurde mit einem Virus verseucht.
Was passiert jetzt? Als erstes wird der Virus überprüfen, ob er bereits auf dem Rechner installiert ist. Wenn nicht, installiert er jetzt seinen virulenten Teil resident im Speicher des Rechners. Als nächstes wird er überprüfen, ob er seinen Minenteil aktivieren soll (dieser Auslöser kann ein Datum, eine bestimmte Datei oder sonst irgendetwas sein).
Dieser Teil des Virus begeht meist ähnliche Gemeinheiten wie das weiter oben abgehandelte Trojanische Pferd. Wenn der Auslöserreiz des Virus nicht gegeben ist, wird der Virus versuchen, sich weiterzuvermehren. Dies geschieht, indem er auf den Datenträgern nach ausführbaren Dateien (EXE oder COM) sucht und sich in diese hineinkopiert (aus dem oben genannten Grund wird auf eine detaillierte Beschreibung des Ablaufs verzichtet). Hat auch das funktioniert, wird das eigentliche Programm gestartet, und der Benutzer des Rechners bekommt bis zu dem Augenblick, in dem der Minenteil des Virus aktiviert wird, nichts davon mit, daß sein Rechner mit einem Virus verseucht ist.

Wie man sieht, sind Viren in zweierlei Hinsicht eine sehr gefährliche Angelegenheit: Erstens können sie, wie ein Trojanerprogramm, großen Schaden anrichten, und zweitens können sie andere Programme infizieren und, wenn diese Programme auch auf anderen Rechnern laufen, diese Rechner ebenfalls verseuchen.

Das beliebteste Programm, das Viren versuchen zu verseuchen, ist COMMAND.COM. Der Grund ist offensichtlich: Ohne COMMAND läuft auf einem normalen PC nichts. Obendrein ist es dem Virus möglich, soviel andere Programme zu erreichen wie irgend möglich, da COMMAND immer als eines der ersten Programme im Speicher ist.

An Nummer zwei stehen Programme in der AUTOEXEC Datei. Da diese bei jedem Start des Rechners ausgeführt werden, kann der Virus sich auch hier zu einem der frühestmöglichen Zeitpunkte im Speicher des Rechners breitmachen.

Die auf anderen Rechnersystemen, wie Amiga und Atari, so verbreiteten Bootsektor-Viren haben sich mittlerweile auch auf dem PC angesiedelt. Die bekanntere Art von Viren auf PCs sind die oben beschriebenen Link-Viren, die sich an andere Programme anhängen. Dagegen kommen die auf Großrechnern bekannten Würmer, die sich über Netzwerke fortpflanzen, auf dem einzelnen PC nicht vor.

Digitales Kammerjägerln

Nach all den Gefahren, die auf den harmlosen Computerbesitzer lauern, nun einige Tips, wie man sich gegen solche Programme schützen kann.
Als erste Maßnahme ist eine garantiert virenfreie Boot-Diskette zu nennen. Diese sollte selbstverständlich schreibgeschützt sein und dazu benutzt werden, das System virenfrei hochfahren zu können, wenn der Verdacht besteht, sich so ein Tierchen eingefangen zu haben. Zum Schreibschutz einer Diskette sei bemerkt: Es gibt ohne Hardwareeingriffe keine Möglichkeit, ein IBM-Laufwerk zu überreden, auf eine

schreibgeschützte Diskette zu schreiben (ganz im Gegensatz zu manch anderen Rechnertypen). Eine schreibgeschütze Diskette ist also nicht durch Viren oder Trojanern angreifbar.

Ein kleiner Tip, um eventuelle Bootsektorviren zu entlarven: Lesen Sie mit einem Diskettenmonitor den Bootsektor der garantiert freien Original-System Diskette aus. Wenn Sie diesen als ASCII-Datei sichern, können Sie die Inhalte anderer Bootsektoren mit dem Original vergleichen. Ist kein Virus vorhanden, müßten die Inhalte der Dateien gleich sein, sofern die Diskettenformate dieselben sind. Diese umständliche Methode ist notwendig, weil der Befehl DISKCOMP, der ja eigentlich Disketteninhalte vergleichen sollte, Bootsektoren nicht in seine Kontrolle einbezieht. Bei MS-DOS 4.00 enthält die mit diesem System erstellte Datei einen weiteren Unterschied, auch wenn die Diskettenformate gleich sind. Das liegt daran, daß ab dieser DOS-Version jede Diskette nach dem Formatieren eine Seriennummer bekommt, die dann natürlich von Diskette zu Diskette verschieden ist. Diese genau vier Byte große Longinteger-Zahl (eine achtziffrige Kennummer) muß beim Vergleich mit beachtet werden.

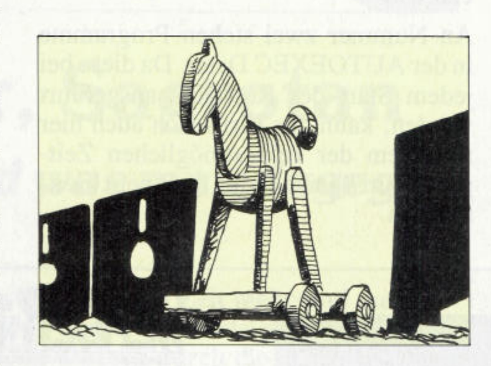
Arbeiten Sie vor allen Dingen niemals mit ihren Orginaldisketten, oder, wenn es sich denn absolut nicht vermeiden läßt (Kopierschutz), schützen sie diese gegen Schreibzugriffe.

Überlegen sie, ob die Größe des Programms wirklich dem entspricht, was in der Dokumentation versprochen wird. Da heutzutage fast niemand mehr komplette Programme in Assembler programmiert, ist der Super-Hyper-Directorylister mit Mausbedienung und Windowtechnik, Programmgröße 1456 Byte, ziemlich unwahrscheinlich. Allein die Laufzeitbibliothek der meisten Hochsprachen hat, bei einem Programm, das gar nichts tut, diese Größe. Überprüfen sie dann und wann die Größe ihrer Programme. Manche Viren ändern nicht die Directory-Einträge des DOS und lassen sich so, durch die Vergrößerung der infizierten Programme, feststellen.

Da besonders Festplattenbesitzer gefährdet sind, einen Rat an diese: Führen Sie regelmäßig Datensicherungen durch! Verwahren Sie aber nicht nur das letzte, sondern die letzten drei Backup-Sets. Bei einem Virenbefall läßt sich der Zeitpunkt des ersten Auftretens meist nicht genau bestimmen.

Überprüfen Sie von Zeit zu Zeit mit CHKDSK Ihre Festplatte. Bestimmte Viren deklarieren die Sektoren mit ihrem Programmcode als unbenutzbar. An und für sich ist eine geringe Anzahl unbrauchbarer Sektoren auf einer Platte normal. Sollte sich die Größe der unbrauchbaren Sektoren der Platte ohne Ihr Zutun einmal ändern, seien Sie alarmiert! Es könnte ein Virus sein.

Der manchmal geäußerte Rat, nur mit kommerzieller Software statt mit PD oder Shareware zu arbeiten, ist zweischneidig. Erstens: Wer hat schon 1000 DM für eine Textverarbeitung übrig? Und zweitens: Auch auf dieser Software kann sich unbeabsichtigt ein Virus einschleichen, wie eine Softwarefirma, die unter anderem Compiler herstellt, unlängst unter Beweis



stellte. Ein von einer Zeitschrift abgedrucktes Virus-Listing (Reden ist Silber, Schweigen ist Gold, oder, wir veröffentlichen ja auch nicht die schweizer Kontonummer unseres Finanzministers) hatte eine derartige Verbreitungsrate, daß es sich auf eine UpDate-Version des vertriebenen Produktes einschleichen konnte.

Aber keine Angst! Es handelte sich dabei um einen Virus der Kategorie "erst Herzinfarkt, dann Genesung" und nicht um einen, der den sofortigen Exitus herbeirief. Außerdem bestätigt dieses Beispiel als berühmte Ausnahme die Regel.

Auch die Angst vor Software, die sich per Virus vor dem Raubkopieren schützen soll, kann getrost abgelegt werden. Hier grasierende Gerüchte konnten bislang nicht bestätigt werden. Abgesehen davon möchte ich nicht in der Haut des Verantwortlichen einer Softwarefirma stecken, die sich auf derartige Weise zu schützen gedenkt; denn sollten durch eine solche Aktion Daten verlorengehen, die nicht in Verbindung mit dem zu schützenden Programm stehen, so liegt ein klarer Fall von Sachbeschädigung vor, und das leistet sich eine Firma nur zweimal: zum ersten- und zum letztenmal.

CMOS-RAM-Viren: Die Entwarnung

Gerade von Besitzern der Amstrad-PCs hört man immer wieder Befürchtungen hinsichtlich des NVRs. Die Angst ist verständlich: Ein Speicher, der selbst nach Abschalten der Gerätes unangetastet bleibt, ist scheinbar prädestiniert für das Ablegen der gefährlichen Kleinstprogramme.

Die Frage nach der möglichen Existenz von Viren im CMOS-RAM ist klar zu beantworten: Es gibt sie nicht! Das NVR kann überschrieben werden, man kann es gegebenenfalls auch löschen, aber da es sich nicht um einen adressierbaren Hauptspeicher handelt, können keine Programme darin ablaufen. Dieser Speicherbereich läßt sich lediglich über I/O-Ports ansprechen und dient zur Aufnahme von Statusinformationen des Rechners. Viren liegen dort so sicher wie in einem Desinfektionsmittel.

Keine Panik, nur Vorsicht

Trotz all dieser Gefahren sollte keiner in größere Panik verfallen. Sicher, hat man sich erst einmal einen Virus eingefangen, so ist das Leid groß. Wer aber mit gebührender Vorsicht an diese Thematik herangeht, der kann sein System auch bei Virusbefall ohne größere Schäden retten. Inwieweit hier die sogenannten Anti-Viren-Programme weiterhelfen, lesen Sie auf den nächsten Seiten.

Tatsache bleibt, daß diese Killerprogramme existieren. Uns bleibt nichts anderes übrig, als der Gefahr mit allen uns zur Verfügung stehenden Mitteln entgegenzutreten. Daß es immer wieder Personen oder Vertriebsfirmen gibt, die hier wenig Verständnis für den Computerbesitzer zeigen, ist ein leidiges Thema. Was etwa im Hirn eines Reiner Becker vorgeht, der mit der Veröffentlichung eines "Virus Construction Set" jedem noch so Computer-Unerfahrenen die Möglichkeit gibt, andere Datenbestände zu zerstören, würde unter den Tatbestand "Beihilfe zur Sachbeschädigung" fallen. Die Menschen sind nun einmal in der Regel keine Engel, und damit hat sich auch eine computerfirme Person abzufinden.

(Robert Haas/jf)



Safer Computing...

Virenschutz-Programme im Test

Die Virenplage hat auf dem PC zwar noch nicht die Ausmaße angenommen, wie auf 'gewissen' anderen Rechnern. Vorbeugen ist jedoch immer besser als neu Installieren. Als Ergänzung zu unserem Beitrag über Viren finden Sie hier nun noch einen Test verschiedener Abwehrmaßnahmen. Aus der Public-Domain stammen die Anti-Viren Tools. Die professionellen Programme, die uns zum Text vorlagen, waren SOFTWAR, Troja-Tools und der Virusdoktor.

Getestet wurden nur Programme, die dem Befall vorbeugen oder ihn aufspüren, nicht jedoch Programme, die infizierte Dateien wieder kurieren. Das Preisspektrum reichte dabei von "fast geschenkt" bis zu für den Hausgebrauch gerade noch zu verkraftenden 150 Mark. Bevor wir die Kandidaten genauer unter die Lupe nehmen, zunächst noch ein kurzer Überblick über verschiedene Maßnahmen zur Vorbeugung:

Methoden der Prophylaxe

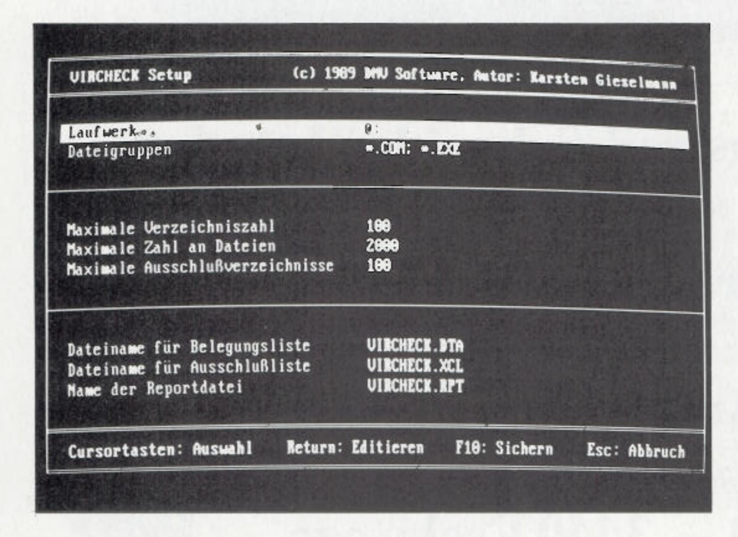
Jeder Virus hinterläßt bei seiner Verbreitung zwangsläufig Spuren auf dem Datenträger, die einen Befall erkennen lassen. Zum einen werden die Programme länger, da sich der Virus meist an das Programm anhängt, zum anderen muß auch der Beginn eines Programms geändert werden, damit der angehängte Virus angesprungen wird. Durch eine ständige Überwachung dieser Werte kann sehr leicht festgestellt werden, wo unerlaubte Änderungen aufgetreten sind. Da man in den MBytes einer Platte sehr schnell den Überblick verliert, vertraut man diese Kontrolle besser eigens dafür geschriebenen Programmen an.
Um sich zu vermehren, muß ein Virus auf den Datenträger schreiben können. Disketten lassen sich per Aufkleber schnell und einfach schützen, bei Festplatten ist das schon etwas schwieriger. Aber auch hier gibt es den Schutz in Form von Software, die die Schreibzugriffe abfängt. In der täglichen Praxis erschweren die meisten Sperren die Arbeit jedoch, so daß sie eher in der Testphase eines neuen Programms hilfreich sind.
Allerdings sei vor Überreaktionen gewarnt: Nicht jede Änderung in einer Datei oder jeder Schreibzugriff eines Programms deutet auf Viren hin. Es gibt Programme, die sich bei einer Installation selbst verändern, beispielsweise CONTEXT, welches seinen Setup im Programm selbst ablegt. Andere Programme können von Zeit zu Zeit völlig legal Daten auf der Platte zwischenspeichern, ohne daß Grund zur Sorge besteht. Die Vorsicht vor Viren sollte also nicht in Paranoia ausarten, ein kühler Kopf ist in jedem Fall zu bewahren. Doch nun zum Test der Programme.

Sicherheit zum Nulltarif?

Stellvertretend für eine Vielzahl von Virenschützern aus der Public Domain soll eine Sammlung aus der Reihe der Deutschen Software Bibliothek vorgestellt werden. Sie enthält fünf Programme mit verschiedenen Schutzfunktionen. DPROTECT sperrt die Schreibzugriffe auf beliebige Datenträger und führt bei ihrem Auftreten einen Neustart des Rechners aus. HDSENTRY fängt ebenfalls alle Schreibversuche auf und warnt den Anwender. (HDSENTRY ist eines der wenigen Programme, die auch den Interrupt 26h abfangen und nicht nur den normalen Interrupt 13h.) Dem Schutz wichtiger Systemdateien ist FLUSHOT gewidmet, welches Zugriffe auf Dateien wie COMMAND.COM oder MSDOS.SYS unterbindet. Ergänzend sind noch die Programme STRING und CHK4BOMB vorhanden, die eine Datei nach ASCII-Texten durchsuchen und diese anzeigen. Darüber hinaus prüft CHK4BOMB, ob sich in einer Datei Programmteile befinden, die Schreibzugriffe enthalten und warnt bei Bedarf. Es ist jedoch fraglich, ob sich dadurch tatsächlich Viren aufspüren lassen, da diese doch meist in irgendeiner Form kodiert und somit unsichtbar sind – aber Anschauen kostet ja nichts. Abgerundet wird die Sammlung durch Textbeiträge über Viren und verdächtige Programme aus der Public Domain. Zwar decken die Programme nicht alle Aspekte der Virenbekämpfung ab, insbesondere HDSENTRY sollte jedoch als effektiver Schutz für die Festplatte in keiner Sammlung fehlen.

Summa Sumarum

Das Programm SOFTWAR spürt einen möglichen Virenbefall durch die Überwachung der Eigenschaften von Dateien auf. Überwacht werden dabei die Einträge für Größe, Datum und Uhrzeit sowie die Dateiattribute. Weiterhin bildet SOFTWAR eine Prüfsumme über die kompletten Dateien und kann somit auch Viren erkennen, die gezielt im Inneren eines Programms wüten. Festgestellte Änderungen werden in einer Datei protokolliert, aus der die möglicherweise befallenen Programme und die an ihnen vorgenommenen Untaten ersichtlich sind. Standardmäßig überwacht SOFTWAR alle Dateien der Typen COM, EXE, SYS und BAT, zusätzlich können Listen mit zu überprüfenden oder von der Prüfung auszunehmenden Programmen erstellt werden. Beim Aufruf des Programms müssen fünf Parameter übergeben werden, was die Bedienung etwas umständlich macht, jedoch größere Flexibilität und Schutz vor Sabotage bieten soll. Die Dauer der Überprüfung hängt von der verwendeten Hardware und der Anzahl



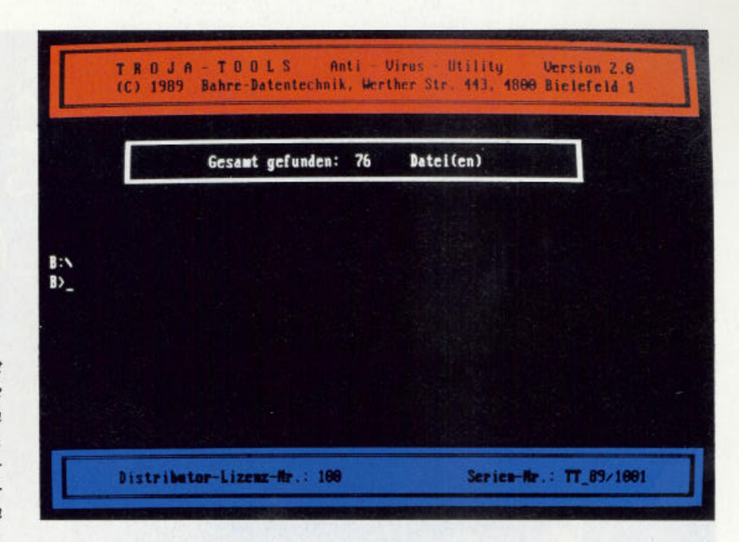
Eine vorbildliche Zusammenstellung von Virenschutzmaßnahmen bietet der Virusdoktor



und Größe der zu überwachenden Dateien ab, ist im Normalfall in wenigen Minuten abgelaufen, so daß ein täglicher Einsatz möglich ist. SOFTWAR überstand alle Tests zur Zufriedenheit und kann als solides Produkt bezeichnet werden, wenngleich der Preis für die Sicherheit etwas zu hoch angesetzt ist.

Grieche mit Problemen

Ähnliche Wege geht das Programm TROJA, welches entweder als Einzelprodukt oder in Verbindung mit anderen Hilfsprogrammen als TROJA TOOLS erhältlich ist. Auch hier werden die Datei-Eigenschaften überwacht, die Prüfsumme wird allerdings nur über den Beginn einer Datei gebildet, was normalerweise für die Erkennung eines Virenbefalls ausreicht. Geänderte Programme werden protokolliert und durch Umbenennung isoliert. Leider erweist sich TROJA als zu unflexibel, neu hinzugekommene oder gelöschte Dateien führen zum Abbruch und erfordern eine neue Initialisierung. Bei dieser wird der gesamte Bestand als 'sauber' betrachtet, so daß Veränderungen, die sich hinter dem Verursacher des Abbruchs befinden, nicht mehr erkannt und zwangsweise legalisiert werden. Auch darf die Reihenfolge der Dateien im Directory nicht geändert werden (beispielsweise durch DiskSort), da auch dies ein Grund für einen Abbruch ist. Im Vergleich zu SOFTWAR, das an diesen Punkten nicht scheitert, muß TROJA leider als unpraktisch gewertet werden.
Dadurch sinkt auch der Wert der anderen Teile von TROJA TOOLS. Diese bestehen zunächst aus einigen Schreibsperren für die Festplatte. Die eine verhindert das Formatieren von Festplatten, mit der anderen können die Zugriffsberechtigungen für ein Laufwerk festgelegt werden, wobei nicht nur Schreib- und Formatierzugriffe gesperrt werden können, sondern auch das ganze Laufwerk aus dem System genommen werden kann. Ergänzt wird das Paket durch ein Map-Utility zur Anzeige residenter Programme (leider ohne die sonst übliche Anzeige der verbogenen Zeiger), ein Verschlüsselungsprogramm für Dateien und eine etwas umständlich zu bedienende Oberfläche für DOS und TROJA TOOLS. Aus dem gesamten Paket konnte lediglich die Laufwerkssperre überzeugen, so daß die TROJA TOOLS ihr Geld im wesentlichen nicht wert sind.



Troja-Tools bietet zwar eine umfassende Ansammlung von Schutzprogrammen, konnte im Test allerdings nicht überzeugen

Komplette Arztpraxis

Ein weiteres Komplettpaket zur Virenbekämpfung ist der VIRUS DOKTOR, der ein breites Spektrum an Schutzmöglichkeiten liefert. Da ist zunächst der obligatorische Schreibschutz, zwar nur mit der Überwachung des Interrupts 13h, dafür jedoch kurzfristig abschaltbar: Im Vergleich zu anderen Sperren kann man den Schutz abschalten, ohne gleich den Rechner neu zu booten, wenn man legal Daten ablegen möchte. Dadurch ist dieses Programm auch während der täglichen Arbeit und nicht nur für den Test von neuen Programmen tauglich. Eine andere Form des Schutzes ist ebenfalls möglich: Das Programm SPERRTSR führt Schreibzugriffe nicht real aus, liefert dem schreibenden Programm aber auch keine Fehlermeldung, so daß das Programm scheinbar normal abläuft. Im Hintergrund werden die Zugriffe protokolliert und können nach dem Programmende untersucht werden. Dritter im Bunde ist SPERRINT, mit dem das Zurücksetzen des Read-Only-Attributs verhindert wird. Die mit diesem Attribut geschützten Dateien können nicht modifiziert werden. Wird zu eben diesem Versuch das Attribut durch einen Virus geändert, schlägt SPERRINT Alarm. Dieser Schutz wirkt effektiv gegen die Ausbreitung von Viren und behindert die normale Arbeit nur wenig. Das Setzen des Attributs für alle Programme ist mit DOS-Tools wie QDos oder XTree kein großer Aufwand, als Nebeneffekt schützt man sich dadurch auch vor eigener Unachtsamkeit. Für den Fall der Fälle bietet der VIRUS DOKTOR auch das schon bekannte Protokollprogramm. Es bildet die Prüfsummen nur über den Dateibeginn und ist somit sehr schnell und dennoch sicher, auch die übrigen Eigenschaften werden überwacht. Dabei stolpert VIRCHEK jedoch nicht über neue oder verschobene Einträge, so daß im Gegensatz zu TROJA ein Betrieb ohne Probleme möglich ist.
Das letzte Programm des Pakets ist ein Backup-Programm für das CMOS-RAM der AT-Rechner. Es dient weniger zur Virenbekämpfung, sondern der Sicherheit bei eventuellen Ausfällen, da mit ihm die Konfiguration schnell wieder hergestellt werden kann. Vorbildlich beim VIRUS DOKTOR auch das Handbuch, welches nicht nur die Programme beschreibt, sondern auch eine gute Einführung in Sachen Virologie liefert.

Bezugsquellen:

Anti-Viren-Tools aus der Public Domain (Sammeldiskette ca. 20 DM):
Deutsche Software Bibliothek
Verlag Nane Jürgensen
Ysenburgstr. 10
8000 München 19

SOFTWAR (148 DM):
SOS Software Service GmbH
Alter Postweg 101
8900 Augsburg

TROJA (98 DM), TROJA-TOOLS (149 DM):
Bahre-Datentechnik
Werther Str. 443
4800 Bielefeld 1

VIRUSDOKTOR (98 DM):
DMV GmbH
Fuldaer Str. 6
3440 Eschwege

(Michael Anton/jf)



Deluxe Paint II Enhanced

Grafiken vom allerfeinsten

Mit Deluxe Paint II Enhanced erschien nun der mit großer Neugier erwartete Nachfolger des weit verbreiteten Deluxe Paint. An eine Update-Version eines so leistungsstarken Programms werden von vorneherein hohe Erwartungen gestellt. Wir haben den Nachfolger für Sie unter die Lupe genommen.

Daß bei Deluxe Paint II enhanced weniger gekleckert als geklotzt wurde, wird schon bei der Installation deutlich. Das Kopieren und Einrichten der vier mit Daten vollgepackten Disketten, dauerte selbst auf dem Amstrad 386 seine Zeit. Danach steht Ihnen der Leistungsumfang des Grafikprogramms fast in vollem Maße zur Verfügung. Fast deshalb, weil ein Zusatzprogramm für die Erstellung von farbigen Zeichensätzen in jedem Fall vor Gebrauch gestartet werden sollte.

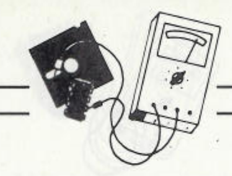
Deluxe Paint benötigt eine Maus. Da das Programm sich nicht sofort über die Existenz einer solchen informiert, ist es empfehlenswert eine Batchdatei einzurichten, die alle für das Programm benötigten Dateien für Sie lädt, sonst sind vorzeitige Programmabbrüche an der Tagesordnung.

Das Auswahlmenü des Grafikmodus ist im Umfang erheblich gewachsen. Es werden nun auch Modi unterstützt, die jenseits der gängigen VGA-Modi liegen. Die Palette reicht von der 640x480 Auflösung bei 256 Farben bis zu 1024x768 Pixel bei 16 Farben. Voraussetzung für derartige Genüsse sind natürlich entsprechendes leistende VGA-Karten, deren genaue Anpassung Deluxe Paint nach Auswahl im Eingangsmenü vornimmt.

Außer den Erweiterungen bezüglich neuer VGA-Karten unterstützt das Programm nun auch EMS-Speicher. Das hört sich allerdings besser an als es sich in der Praxis herausstellt. Entnimmt man dem Bild einen bestimmten Bereich, um diesen gesondert zu bearbeiten, so sollte man erwarten, daß bei der Bearbeitung die Veränderungen gleich bildlich dargestellt werden. Leider wird der Ausschnitt ab einer bestimmten Größe durch einen simplen Rahmen dargestellt, was die Bearbeitung nicht immer unterstützt.

Die Arbeit mit diesen Bildausschnitten, den sogenannten Brushes, steht im Mittelpunkt bei der Erstellung von Grafiken mit Deluxe Paint II enhanced. Hier zeigt sich dann auch am besten die Möglichkeitenvielfalt, mit der sich das Programm darstellt.

Ausschnitte können rechteckig oder auch in beliebig anderer Form dem Bild entnommen werden, wobei letztere Funktion nicht durch eine besondere Benutzerfreundlichkeit glänzt. Ein solcher Auschnitt wird vom Programm nur dann anerkannt, wenn er exakt abgeschlossen wird. Das heißt, daß der Benutzer mit dem Mauszeiger eine Fläche definieren muß, deren Anfangs- und Endpunkt übereinstimmen müs-



sen, ein Unterfangen, das mitunter zum Glücksspiel werden kann.

Ist allerdings erst einmal eine Brush definiert, so kann man diese nach Belieben bearbeiten. Dehnen, Stauchen und Drehen sind selbstverständliche Standardoperationen. Dazu kommen Extras wie transparente und schattenhafte Darstellung oder auch Versetzen bei Beibehaltung vorher definierter Hintergrundfarben.

Das räumliche Verschieben und Drehen der Auschnitte ist im Vergleich zum Vorgänger erheblich komfortabler geworden. Einige Sonderfunktionen wie das Projezieren auf vorher bestimmte geometrische Figuren läßt das Programm allerdings vermissen.

Im äußeren Erscheinungsbild hat sich im Vergleich zum Vorgänger auf den ersten Blick wenig geändert. Eine Menüleiste am rechten, sowie ein Pulldownmenü am oberen Bildschirmrand dienen der Funktionswahl. Gerade die Icons in der sogenannten Toolbox fallen durch zahlreiche Unterfunktionen auf.

Einmal mehr kann gutem Gewissens behauptet werden, daß die genaue Auflistung aller Funktionen, die Deluxe Paint unterstützt, den Rahmen dieses Artikels sprengen würde. Einen Einblick in die Vielfalt soll Ihnen unsere Abbildung geben.

Neu ist auch die Möglichkeit, Bilder zu bearbeiten, deren Umfang den des normalen Bildschirms sprengen. Wenn man diese Werke über den Bildschirm verschiebt, so macht sich die Nutzung zusätzlichen Speichers vorteilhafter bemerkbar als bei der Ausschnittbearbeitung. Zügig wird die Bildfläche hin- und hergeschoben. Selbstredend ist der Benutzer in der Lage, sich die verkleinerte Version des ganzen Bildes im Überblick anzusehen.

Eine ähnliche Funktion steht bei der Auswahl von Ausschnitten oder Bildern aus dem Directory zur Verfügung. Eine verkleinerte Version des auf Datenträgers befindlichen Bildes wird eingeblendet. So wird ein komfortables Durchsuchen des Disketteninhalts ermöglicht.

Die erweiterte Version von Deluxe Paint wurde auf eine große Anzahl neuer Drucker angepaßt. Weit über 200 Geräte, darunter diverse Farbdrucker, werden unterstützt.

Deluxe Paint II enhanced bietet außer dem eigentlichen Grafikprogramm einige Utilities, die weiteren Komfort verantworten. Das Programm Camera ist schon vom Vorgänger bekannt. Das residente Programm sichert auf Tastendruck den aktuellen Bildschirm und ermöglicht so eine Bearbeitung beliebiger Screens mit Deluxe Paint.

Ein weiteres sinnvolles Utility verbirgt sich hinter dem Namen Convert. Mit Hilfe dieses Programms lassen sich von anderen Grafikprogrammen gesicherte Bilder in ein für Deluxe Paint akzeptables Format verwandeln. Mit TIFF, MacPaint, Colorix oder MS Windows Paint 2.0 erstellte Grafiken können konvertiert werden. Das Programm funktioniert in beiden Richtungen, was heißen soll, daß auch mit Deluxe Paint II gemalte Bilder in die angegebenen Fremdformate verwandelt werden können.

Gallery nennt sich ein weiteres Hilfsprogramm, mit dem Bildpräsentationen arrangiert werden können. Dabei werden die Bilder nicht einfach ein- und ausgeblendet. Dem Benutzer stehen vielmehr eine Reihe von Überblendungsmöglichkeiten zur Auswahl.

Ein fast perfektes Produkt

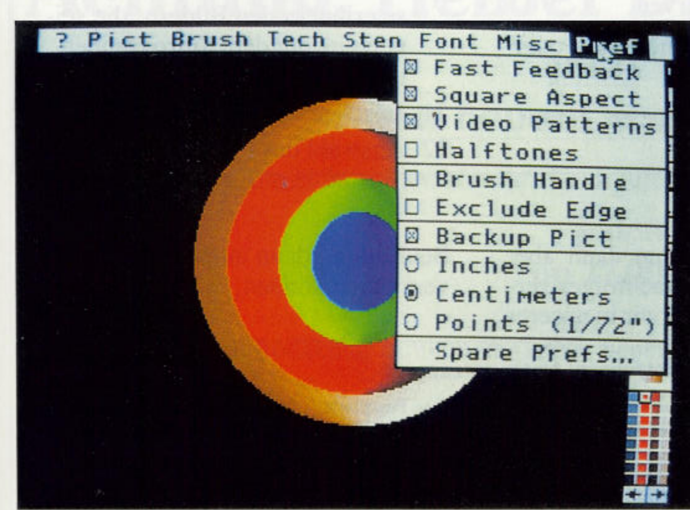
Sicherlich wird auch Deluxe Paint II enhanced aus einer grafisch völlig unbegabten Person keinen Künstler machen, aber selbst dieser Personengruppe wird nach wenigen Minuten Erstaunliches gelingen.

Die Anwenderfreundlichkeit und die Menge von Features weiß den Benutzer nicht nur am Anfang zu überwältigen.

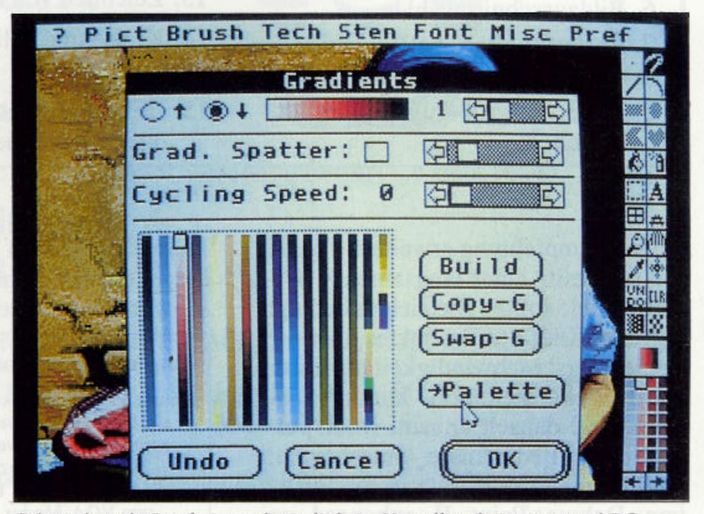
Daß es sich letztlich dennoch nur um ein fast perfektes Produkt handelt, liegt an einigen Schwächen, die nicht als Nebensächlichkeit abgehandelt werden sollte. Die noch nicht ganz ausgereifte Nutzung des EMS-Speichers gehört dazu.

Besitzer einer Hercules-Karte werden auf der einen Seite positiv überrascht sein, mit welcher Präzision ihr Grafikmodus von diesem Programm unterstützt wird. Auf der anderen Seite müssen sie sich bei der Lektüre des Handbuches stark vernachlässigt fühlen, geht dieses doch nur unzulänglich auf Spezialitäten dieser Grafikkarte ein. Weiterhin ist zu kritisieren, daß die Einstellung der Durchlaufgeschwindigkeiten der Farben bei manchem Grafikmodus nicht unproblematisch ist. So besteht im hochauflösenden EGA-Modus bei der neuen Amstrad-PC-Reihe akute Absturzgefahr, wählt man hier eine hohe Geschwindigkeit.

Das Programm empfiehlt dem “Maler“, er solle doch einen Rechner mit einem 286er- oder gar 386er-Prozessor benutzen. Man kann sich diesem Aufruf nur anschließen und ihn um eine



Die thematisch orientierten Pulldownmenüs garantieren eine problemlose Bedienung



Schwachpunkt Laufzeitgeschwindigkeit. Vor allem beim Amstrad PC herrscht hier bei hohen Cyclen Absturzgefahr



Deluxe Paint II enhanced

1. Form und Größe des Pinsels
2. Linien in allen Varianten
3. Rechtecke und Quadrate, gefüllt oder als Rahmen
4. Vielecke, gefüllt oder als Rahmen
5. Fill-Funktion
6. Bildausschnitt wählen
7. Rasterfunktion, beliebige Größen einstellbar
8. Lupe, Vergrößerung: 2- bis 16fach
9. Direktentnahme der Farben für Vorder- und Hintergrund
10. Rücknahme der letzten Funktion
11. Anwahl der vorher definierten Farbverläufe
12. Art des Pinselstrichs
13. Zeichnen von Kurven
14. Kreise und Ellipsen, gefüllt oder als Rahmen
15. frei gezeichnete Formen, gefüllt oder als Rahmen
16. Spraydose, Durchmesser der Sprühfläche kann eingestellt werden
17. Wahl der Schriftart
18. Perspektivische Verzerrungen
19. Bildfläche scrollen
20. symetrisches Zeichen (Ergebnisse erinnern den Bildern, die seinerzeits mit dem Spirographen erstellt wurden)
21. Löschen des Bildschirms
22. Auswahl des Füllmusters

weitere Empfehlung erweitern. Wenn Sie im Besitz einer langsamen VGA-Karte sind, sollten Sie sich unbedingt vor dem Kauf des Produktes über die Bedienungsgeschwindigkeit auf Ihrer Hardwarekonstellation kundig machen, und danach anhand Ihrer persönlichen Bedürfnisse entscheiden, ob Sie geneigt sind, sich in den Besitz von Deluxe Paint II enhanced zu bringen.

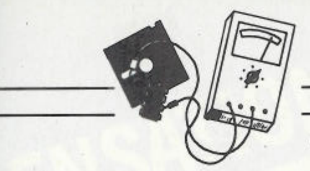
Diese Schwächen sollten dann aber bei einem möglichen Nachfolger verschwunden sein, so daß wir in unserer Beurteilung dann auch auf das "fast" verzichten können.

(jf)

Empfehlungen, Voraussetzungen, Unterstützungen:

Deluxe Paint II enhanced unterstützt:
- alle Grafikkarten (auch erweiterte VGA-Modi)
- empfiehlt die Verwendung von:
  - 286- oder 386-Prozessor
  - 512 kByte EMS bei Nutzung von erweitertem VGA-Modus
- Festplatte
- 640 kByte Arbeitsspeicher
- MS-DOS 2.1 (oder höher)
- Microsoft kompatible Maus

Deluxe Paint II enhanced ist nicht kopiergeschützt.
Hersteller: Electronic Arts



The Manhole

Ein Kinderspiel

Tatsache ist, daß das Alter, in dem eine Person zum ersten Male in Berührung mit einem Computer tritt, stetig sinkt. So entsteht mittlerweile Software, die in starkem Maße auf Kinder abzielt. The Manhole ist ein solches Programm. Phantasie ist in diesem Spiel Trumpf.

Hier geht es einmal nicht darum, die übliche Prinzessin aus den Klauen des üblichen Drachen zu retten. Das Entdecken einer Phantasiewelt steht im Vordergrund.

Alles beginnt vor einem Kanaldeckel, aus dem ein riesiger Baum in den Himmel wächst. Klettert man an diesem hinauf, so gelangt man durch ein Loch in der Wolkendecke in eine nächtliche Gegend, in deren Zentrum ein alter Burgturm steht. Von hier aus kann die Reise weitergehen zu einem freundlichen Drachen (nicht der übliche!), einem Elefanten, mit dem man per Gondel ein Kanalsystem erforschen kann oder einem überaus zuvorkommenden Walroß. Man besucht den Meeresboden oder findet sich in der Teetasse eines alten Hasen wieder.

In Manhole ist alles möglich. In den blendend gezeichneten Bildern können mit der Maus oder dem Joystick eine Unzahl von Einzelheiten angeklickt werden, was fast immer zur Folge hat, daß sich irgendetwas auf dem Bildschirm tut. Da wird das Szenario aus einem anderen Blickwinkel gezeigt, Dinge beginnen sich zu bewegen oder geben Laute von sich. So wird das Spiel zu einer spannenden Reise durch eine nette Märchenwelt.

Auf der Reise durch die Fantasiewelt trifft man auf allerlei Fabelwesen

...und Erwachsene ebenso

Manhole ist ein fast perfektes Kinderspiel. Das zeigt sich speziell daran, daß es in besonderem Maße vor allem Erwachsene anspricht. Ohnehin weiß das Programm seine besonderen Stärken genau dann auszuspielen, wenn sich Eltern und Kinder gemeinsam vor den Bildschirm begeben.

Schade ist, daß Manhole komplett in Englisch gehalten wurde. Das stimmt besonders traurig, da viele der anzutreffenden Märchengestalten per Sprachausgabe zu reden beginnen, wenn sie angeklickt werden.

Programmtechnisch fiel nur ein Mangel auf: Die erwähnte Sprachausgabe wurde nicht prozessorunabhängig programmiert, so daß die Stimmen etwa bei einem 386er "mickymausige" Formen annehmen.

(jf)

Hersteller: Activision
Preis: Stand bei Redaktionsschluß noch nicht fest
Besonderheiten: Das Programm ist nicht kopiergeschützt, benötigt mindestens 512 kByte RAM und einen Farbmonitor und unterstützt Mouse, Joystick; alle gängigen Grafikkarten inklusive MCGA, sowie Soundkarten von Ad-Lib bis zu Roland MT-32.

Achtung: Heißer Draht

jetzt

mit neuer Durchwahl!



Haben Sie Fragen, die nicht warten können? Dann rufen Sie uns an! Unser Leser-Telefonservice steht Ihnen

jeden Mittwoch
von 17.00 bis 20.00 Uhr

zur Verfügung.

Sie erreichen Ihren Redakteur direkt unter seiner neuen Durchwahlnummer. Auf Ihren Anruf freuen sich:

Claus Daschner (CPC)
(0 56 51) 809–750

Joachim Freiburg (CPC/PC)
(0 56 51) 809–751

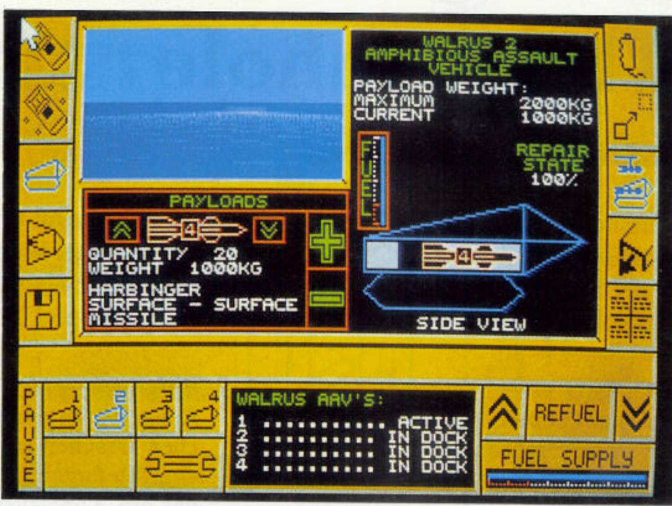
Ralf Schößler (PCW)
(0 56 51) 809–752

Die in früheren Heften veröffentlichten Telefonnummern sind ab Januar 1990 ungültig – diese Nummern bitte nicht mehr anwählen!

Carrier Command

Komplexe Strategie-Simulation

Die Erde tut sich auf und wenn sie das tut, dann kommt meist nichts Gutes heraus: brennende Lava. Wenn diese Art von Ausbrüchen unter Wasser stattfindet, dann bilden sich häufig neue Landmassen. Bei einem der letzten größeren Ausbrüche entstand gleich eine ganze Inselgruppe, um die sich zwei gegnerische Parteien sogleich zu streiten begannen. Diese Parteien sind der Spieler und sein digitales Gegenüber.



Einen grafischen Leckerbissen bietet Carrier Command allen Action- und Strategiefreunden

In Carrier Command bekommen Sie einen kompletten Flugzeugträger gestellt, der mit allen Extras ausgerüstet ist, die man zum Nutzbarmachen von Neuland benötigt. Er ist mit Flugzeugen und Amphibienfahrzeugen ausgestattet, die sowohl zum Kampf als auch zur sinnvollen Anwendung genutzt werden können.

Vernetzung der Inseln

Ihre Aufgabe ist es nun, so viele Inseln wie möglich zu vernetzen, um damit einen Produktions- und Verteidigungsverband zu errichten. Dazu wird Insel für Insel angesteuert. Mittels Amphibienfahrzeug wird eine Robotereinheit auf das neue Land gebracht, die wahlweise Rohstoffe fördert, mit der Produktion diverser Gebrauchsgüter beginnt oder einen Verteidigungsstützpunkt errichtet.

Über die Produktionsinseln beziehen Sie Ihren Nachschub an Treibstoff, Waffen und anderen Materialien. Dieses ist auch bitter notwendig, weil spätestens bei der ersten Konfrontation mit Ihrem Computergegner, der ebenfalls das neue Land zu seinen Gunsten ausschlachten will, das Programm einer Materialschlacht nahekommt.

In diesen Situationen hat der Spieler das komplette Kampfsystem des Carriers zu bedienen, was alles andere als einfach ist. Das hat nichts mit fehlender Übersichtlichkeit zu tun ganz im Gegenteil: Durch ein sehr gut nachvollziehbares Icon-System sind die diversen Aktionen schnell durchführbar. Das Problem ist vielmehr, daß der Spieler jedes einzelne Waffensystem separat bedienen muß. Während dieser Zeit sind die anderen Systeme dem Gegner ausgeliefert, der seinerseits mit seinen Truppen parallele Angriffe führt. So müssen die Kämpfe sehr taktisch geführt werden, um sie überleben zu können.

Carrier Command läßt sich auf zwei verschiedene Weisen spielen: als Action- oder als Strategiespiel. Unterschiede liegen in den Startbedingungen: Beim Strategiespiel beginnen die Kontrahenten, weit voneinander entfernt, die Inseln unter eigene Kontrolle zu bringen. Die Action-Version startet in einem vorangeschrittenen Stadium. Außerdem befindet sich der Carrier gleich vor der Küste einer feindlichen Insel, was mit großer Wahrscheinlichkeit zu einem sofortigen Spielende führt, wenn man die Steuerung nicht sicher beherrscht.

Die grafische Präsentation ist in keinem Punkt zu bemängeln. Ob Sie sich nun im Träger selbst befinden, einen Jäger fliegen oder mittels Amphibienfahrzeug die Meere befahren: Übersichtliche Displays, schnell und ruckfrei animierte Ausblicke und problemlose Ansteuerung der einzelnen Funktionen sind in jeder Phase gewährleistet.

Gerade der letzte Punkt ist aufgrund der zum Teil hektisch werdenden Spielsituation besonders zu begrüßen. Das liegt vor allem an der Größe der Icons, die nicht pixelgenau getroffen werden müssen.

Nach so vielen Pluspunkten soll nicht verschwiegen werden, daß auch bei Carrier Command nicht alles Gold ist, was glänzt. Bei all der fantastischen Präsentation kommt der Spielwitz letztlich ein wenig zu kurz. Nach mehreren Stunden des Fliegens oder Schwimmens wiederholen sich die eigentlichen Tätigkeiten während des Spiels doch in erheblichem Maße. Daß das Programm dann doch der Langeweile entwischt, liegt an der besagten Aufmachung.

Ein weiterer Nachteil ist die Möglichkeit des Schummelns, falls der Spieler in bestimmten Aktionen fehlerhaft reagiert hat. Sichert man das Spiel in regelmäßigen Abständen, so kann man nachteilig erspielte Situationen ohne Probleme solange durchspielen, bis es letztlich zu einem für den Spieler gutem Ende kommt. Das liegt daran, daß Carrier Command nach dem Laden eines Spielstandes den Verlauf nicht variiert, sondern eine einmal gewählte Taktik stetig wiederholt.

Brillant gemacht

Man kann zu Kampfspielen stehen wie man will, manchmal ist ein solches Programm durch seine perfekte Machart imstande, auch den größten Skeptiker dieses Genres zu überzeugen. So geschehen bei Carrier Command. Die Action-Variante halte ich für wenig gelungen, da die Überlebenschance stark gen Nullpunkt strebt. Die Strategie-Version dagegen gehört zu dem besten, was mir seit langem an PC-Spielen untergekommen ist. Brillante Animationen der gefüllten Vektorgrafiken und komplexe gegenseitige Beeinflußungen der einzelnen Aktionen sind in der Lage, den Spieler lange Zeit am Bildschirm festzuhalten. Was will man mehr?

(jf)

Hersteller: Rainbird
Preis: 89 DM
Besonderheiten: unterstützt alle gängigen Grafikmodi und ist durch Kopierschutz und Security-Abfrage überflüssig oft geschützt.



Kühles Wetter, heiße Preise

Sonderhefte Joyce Nr. 2, 3, 4

als Paket für sagenhafte 29,– DM*

DMV braucht Platz für neue Ideen – darum müssen wir unsere Lager räumen.

Allen JOYCE-Besitzern machen wir deshalb ein einmaliges Angebot:
JOYCE Sonderhefte 2, 3 und 4 sind ab sofort im Kombi-Pack zum Preis von 29,– DM beim Verlag zu beziehen.
– Sie sparen mehr als 50% gegenüber dem Einzelpreis!

JOYCE Sonderhefte sind Sonderpublikationen der PC AMSTRAD International und bieten jeweils auf 120 Seiten ausschließlich erstmalig veröffentlichte Beiträge, Tips und Tricks zu PCW 8256, 8512 und 9512.



Aus dem Inhalt
JOYCE Sonderheft 2:
Adreßverwaltung
Archivprogramm
(Video-oder Literaturverwaltung)
PASCAL-Compiler in BASIC
Suburbia (Spiel ähnlich Monopoly (R))
Turbo-PASCAL-Grafikroutinen ohne GSX
Komfortable Balkengrafik
JOYCE-Zweitlaufwerk selbst anschließen
3-D-Plotter
Etikettendruckprogramm
Ordnung auf der Diskette mit LocoScript
Funktionstasten selbst belegen
Spaltensatz unter LocoScript
dBase-Handbuch selbst ausdrucken
LOGO- Funktionenzeichner

Aus dem Inhalt
JOYCE Sonderheft 3:
Vokabeltrainer
RAM-Monitor – Speicherinhalte verändern
Memory-Spiel
Mini-DTP-Programm
Drucker-Spooler unter CP/M
Disketten-Kopierprogramm bis 43 Spuren
Reset ohne Datenverlust
Grafik auf dem JOYCE-Drucker
Tastaturbelegung unter CP/M und LocoScript ändern
Reset ohne Datenverlust
Super-Werkzeugkiste dBase
Grafikutilities für LOGO

Aus dem Inhalt:
JOYCE Sonderheft 4:
Strickmustergenerator
WordStar-Verbesserungen
Bundesligasimulator
Super Reaktionsspiel
FILEMANAGER, Pulldown-Menüs
Stichwortverzeichnis,
Astrologieprogramm
Diskettenmonitor
Hauptstädte raten in LOGO
Statuszeile für dBase und Basic
Hardcopyroutine für 24-Nadler
LOGO macht Schachteln
dBase-Literaturverwaltung
Universelles Werkzeug zur Veränderung von dBase-Dateien

Und dazu die Databoxen aller Sonderhefte im Paket !

5 Disketten Joyce-Power für traumhafte 79,– DM*

* **Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.**





MS-DOS, die ersten Berührungen

Prompt und Dateien



Ab dieser Ausgabe wollen wir eine kleine Serie starten, die wichtige Hinweise und Tips für Einsteiger unter MS-DOS geben soll. Wir wollen Schwierigkeiten beleuchten, die bei den ersten Berührungen mit dieser Betriebssystemebene auftauchen, und Tips und Tricks beschreiben, die wenig oder keine Berücksichtigung in den Handbüchern finden.

Als ich zum ersten Mal einen Personal Computer einschaltete, erwartete ich, geschädigt von der Farbenpracht bunter Werbungsbilder, ein mich verblüffendes Spektakel. Vor meinem geistigen Auge liefen die tollen Demos aus den Schaufenstern ab: Bunte Grafiken, die versuchten, den zukünftigen Benutzer durch überrumpelnde Texte vom Kauf des Gerätes zu überzeugen. Ich hatte Symbolbilder im Sinne, die mir durch Anklicken mit der Maus weitere Tiefen des Computers erschließen würden. Ich dachte an auf- und zublendende Fenster, Tabellen, integrierte Bilder, benutzerfreundliche Eingabemedien und, und und..

Die Enttäuschung war natürlich groß, als sich nichts von all dem präsentierte, sondern der Computer seine Existenz durch einen banalen Buchstaben preisgab.

Diese Enttäuschung nach dem "ersten Mal" wird so mancher neue PC-Besitzer hinter sich haben; denn das Betriebssystem eines PC ist eine trockene Angelegenheit. Man nennt es Betriebssystem, weil erst diese Ansammlung von Systemprogrammen, den allgemeinen Betrieb eines Computers ermöglicht. Das Betriebssystem bildet die Schnittstelle zwischen Gerät und Anwender. Es sorgt dafür, daß die eingegebenen Befehle in eine für den Computer verständliche Sprache übersetzt werden. Dabei sind diese Systemprogramme dafür geschaffen, allgemeine Dienste zu leisten, etwa den Umgang mit den auf den Speichermedien befindlichen Dateien (laden, speichern, umbenennen, löschen etc.).

Der Benutzer kommuniziert mit diesem Programm mittels Eingabe von Befehlen, wobei MS-DOS sich solange im Wartezustand befindet, bis es durch den Druck der ENTER-Taste dazu veranlaßt wird, sich des Befehls anzunehmen. Zuerst wertet das Betriebssystem diese Eingabe aus. Handelt es sich dabei um ein Wort, das dem Computer bekannt ist (DIR, COPY, etc.), so wird ein Kommandointerpreter beauftragt, die Schritte einzuleiten, die zur Ausführung des Befehls notwendig sind; andernfalls erscheint eine Fehlermeldung auf dem Bildschirm.

Alle diese Eingaben werden per Tastatur eingegeben und erscheinen hinter dem schlichten Betriebsbereitschaftszeichen, dem sogenannten Prompt. In der Regel besteht dieser aus der Angabe des aktuellen Laufwerks. Wenn Sie einen Computer mit Festplatte besitzen, so wird dieser sich aller Wahrscheinlichkeit nach mit einem **C:>** melden, ansonsten ist anstelle des Cs der Kennbuchstabe des angesprochenen Laufwerks zu finden.

Variationen über Prompt

Der Benutzer kann das Prompt fast nach Belieben verändern. Dazu steht ihm ein MS-DOS-Befehl zur Verfügung, der zweckmäßigerweise PROMPT heißt. Tippen Sie beispielsweise auf der MS-DOS-Ebene die Zeile

PROMPT Befehl eingeben

ein, so meldet sich der Computer nach jedem nun folgenden Befehl mit der Meldung "Befehl eingeben". Auf diese simple Weise lassen sich alle denkbaren Meldungen als Betriebsbereitschaftszeichen auf den Bildschirm bringen.

Natürlich kann über den Sinn gestritten werden, was ein solches Prompt anbelangt. Sieht man sich allerdings die Liste der von MS-DOS zur Verfügung gestellten Sonderinformationen an, so bemerkt man schnell, daß außer Wortlautmeldungen auch Systeminformationen in das Prompt integriert werden können. Tabelle 1 zeigt eine Auflistung aller Promptzusätze und was sie auf dem Monitor bewirken.

Um diese Zusätze zu nutzen, reicht es aus, dem Befehl Prompt eine beliebige Reihe von nutzbaren Symbolen folgen zu lassen. Sie sehen, daß jeder Informationsabruf aus einem $-Zeichen und einer Buchstabenkennung besteht. Ein Beispiel soll hier zur Erläuterung dienen:

PROMPT dthhhhhh$__ pg

Wie Sie der Tabelle entnehmen können, bewirkt $d, daß das aktuelle Datum im Format Tag.Monat.Jahr geschrieben wird. Es folgt ein Leerzeichen, welches dafür sorgt, daß es nach dem Datum nicht gleich weitergeht. $t schreibt nun die aktuelle Zeit auf den Monitor, und das in einer äußerst präzisen Genauigkeit. Wir wollen uns auf eine minutiöse Genauigkeit beschrän-

Zulässige Prompt-Kommandos:

$b	– Pipesymbol - ASCII 124 "|"
$d	– aktuelles Datum im Format TT.MM.JJ
$e	– Escapezeichen
$g	– Größer-Symbol ">"
$h	– Backspace - löscht das letzte Zeichen
$l	– Kleiner-Symbol "<"
$n	– aktuelles Laufwerk
$p	– aktuelles Verzeichnis im benutzten Laufwerk
$q	– Gleichheitszeichen "="
$t	– aktuelle Zeit im Format SS.MM.SS.HH
$v	– Die MS-DOS-Versionsnummer
$__	– Carrige RETURN - Neue Zeile
$$	– Dollarzeichen "$"

Abbildung 1

JOYCE/PCW
DATABOX





PC 1512/1640
DATABOX



DATABOX ist der preiswerte Software-Service Ihrer Zeitschrift
DATABOX enthält lauffähige Programme für Ihren Einsatz
DATABOX lohnt sich auf jeden Fall - Monat für Monat

JOYCE PCW 8256/8512/9512

JUCA
Der zweite Teil der Serie behandelt die Module "Teilprogramme" und "Anzeigen". Beide sind auf der DATABOX vorhanden. Mit den Datein der DATABOX 12/89 ist JUCA dann bereit, Ihre Daten aufzunehmen.

Tastenspiele
Das Arbeiten mit Steuerzeichen ist oft ein umständliches Unterfangen, ob nur mal kurz der Drucker umgeschaltet oder etwas auf den Bildschirm gebracht werden soll. Tastenspiele ermöglichen die Eingabe der Steuerzeichen direkt von der Tastatur und sorgen so für neuen Komfort.

Bonusprogramm: Quiz-Master
Thema »Künstliche Intelligenz«. Mit dem Programm Quiz-Master stellen wir Ihnen ein lernfähiges Programm zur Verfügung. Anhand bestimmter Kriterien ist dieses Programm in der Lage, beliebige Gegenstände/Personen zu erraten.

PC 1512/1640

Ein Helfer für Batches
Die Unterbindung der lästigen MS-DOS Fehlermeldung über nicht zugriffsbereite Laufwerke besorgt unser Utility-Programm. Die Fehlermeldung wird komfortabel abgefangen und der Bildschirm ordentlich wieder restauriert.

Patzige Datensicherung
Ein fast perfekter Schutz vor Zugriffen nicht berechtigter Personen auf Ihre Datenbestände wird mit der patzigen Datensicherung erreicht. Lassen Sie Ihren Computer "giften"...

Bonusprogramm
Diesen Monat entführen wir Sie in die harte Welt eines jungen dynamischen Geschäftsmann, der sich in den Kopf gesetzt hat, die weite Welt mit einer Flugzeuggesellschaft zu erobern. Airline wird auch Sie aus dem Pilotensitz heben.

Einzelbezugspreis für DATABOX:
PCW - 3-Zoll-Disk. PC - 5 1/4-Zoll-Disk.

24,– DM (Unverbindliche Preisempfehlung)
Wenn Sie über den DMV-Verlag bestellen, gilt folgendes:

Inland:		**Ausland:**	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

Das DATABOX-Abo kostet:
Als Diskette für 1/2 Jahr (6 Lieferungen)

Im Inland und West-Berlin	150,– DM
Im europäischen Ausland	160,– DM
Im außereuropäischen Ausland	180,– DM

Als Diskette für 1 Jahr (12 Lieferungen)

Im Inland und West-Berlin	300,– DM
Im europäischen Ausland	320,– DM
Im außereuropäischen Ausland	360,– DM

Bitte benutzen Sie für Ihre Bestellung die Abo-Karte.

Zahlungshinweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferungen ins Ausland ist Nachnahme nicht möglich.)

Widerrufsrecht:
Jeder Abonnent hat das Recht, seine Bestellung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich zu widerrufen. Die rechtzeitige Absendung des Widerrufschreibens genügt zur Fristwahrung.

Bitte benutzen Sie die Bestellkarte

DMV-Verlag · Postfach 250 · 3440 Eschwege



ken. Da wir nicht in der Lage sind, dies dem Prompt-Befehl direkt beizubringen, löschen wir einfach die letzten sechs Zeichen dieser Zeile. **hhhhhh** erledigt das für uns. Es folgt ein **$__**-Zeichen. Die Tabelle gibt Auskunft darüber, daß nach diesem Zeichen ein neue Zeile begonnen wird. In der erscheint nun durch **$p** das aktuelle Laufwerk und der Verzeichnispfad. Mit **$g** folgt ein Größer-Zeichen, daß nur zur Abtrennung von Prompt und der folgenden Befehlszeile dienen soll.

Haben wir nun diese Befehlszeile in unseren PC eingegeben, so erscheint nach jedem Druck auf die RETURN-Taste eine Meldung der folgenden Art:

MON 27.10.89 11.34
C:\DOS\TOOLS>

Sie sehen, daß durch die Wahl eines solchen Bereitschaftszeichens eine Anzahl von Informationen an den Benutzer weitergegeben werden.

Viele Varianten sind denkbar, nicht zuletzt aufgrund der Kombinationsmöglichkeiten von Realtext und Sonderzeichen. Probieren Sie einmal folgendes:

PROMPT $v am $d. Verzeichnis: $p $__

Nach einigen Experimenten werden Sie Ihr bevorzugtes Prompt auf dem Bildschirm vor sich sehen. Diese Informationen vergißt der Computer allerdings, sobald er ausgeschaltet wird.

Das hat zur Folge, daß wir nach jedem Anschalten unseren Promptbefehl erneut eingeben müssen. Vorteil daran ist, daß wir schnell die Syntax des Prompt auswendig lernen werden, Nachteil ist das immerwährende Eingeben der doch mitunter langen Befehlszeile. Daher wollen wir an dieser Stelle eine sogenannte Stapeldatei erstellen, die uns diese Arbeit abnehmen soll. Lassen Sie uns in dieser Folge bei dem puren Befehl bleiben, der das für Sie erledigt. Was sich genau hinter dem Ausdruck Stapeldateien verbirgt, wie man Sie anlegt, handhabt und geschickt einsetzt, wollen wir im nächsten Heft näher beleuchten. Nun aber zu unserer Promptsicherung: Mit

COPY CON P.BAT
PROMPT $d thhhhhh__ pg
<F6>
<RETURN>

erstellen wir auf dem aktuellen Laufwerk eine Stapeldatei mit dem Namen P.BAT. An dieser Stelle soll aus obigem Grunde als Infornation ausreichen, daß durch bloßes Aufrufen dieser Datei unser Beispielprompt auf Ihrem Computer installiert wird. Rufen Sie diese Datei folgendermaßen auf:

P <Return>

Was eigentlich ist eine Datei?

Jeder, der von Computern redet, benutzt spätestens im dritten Nebensatz das Wort "Datei". Alles, was mit Software zu tun hat, dreht sich offensichtlich um dieses Zauberwort. Selbst wir haben vor ein paar Zeilen eine Datei erstellt und darüber hinaus auch noch aufgerufen. Was genau verbirgt sich aber nun hinter dem Wort "Datei"?

Die einfachste Erklärung des Begriffes liegt im Wort selbst bereits versteckt: Danach ist eine Datei nichts anderes als eine sinnvolle Ansammlung von Daten. Das können nun lauter Befehle einer Programmiersprache sein, die in ein für den Computer lesbares Format gebracht worden sind; dann hätten wir es mit einem Programm, einer Befehlsdatei, zu tun.

Es könnte sich dabei aber auch um eine Aneinanderreihung von MS-DOS-Befehlen handeln, wie etwa um den Befehl, der unser Prompt manipuliert. Einen solchen Stapel von MS-DOS-Befehlen nennen wir Batchdatei.

Dateien, die nichts anderes als Text enthalten (solche, die beispielsweise Ihre Textverarbeitung anlegt) nennt man natürlich Textdateien, andere, deren Aufgabe das Lagern von systeminternen Einstellungen ist, Systemdateien. Man könnte diese Liste fortführen, wir wollen es aber bei diesen wenigen Beispielen belassen.

Wie bereits angedeutet, befinden sich diese Dateien auf den Datenträgern (Disketten, Festplatte). Eine Aufzählung des Inhaltes eines solchen Speichermediums bekommt man mit dem Befehl DIR. Wir geben ein:

DIR <RETURN>

Auf dem Bildschirm erscheint ein genaues Inhaltsverzeichnis des aktuellen Datenträgers. Abbildung 2 zeigt ein mögliches.

Wir sehen eine große Menge von Informationen, die allesamt unsere Dateien betreffen. Jede Datei besteht aus einem Namen und einer Kennung. Der Name darf bis zu acht Zeichen lang sein und sollte aus einer Kombination von Buchstaben und Zahlen zusammengesetzt werden. Nach dieser ersten Zeichenfolge folgt ein Punkt und eine maximal drei Zeichen umfassende Datei-Namenserweiterung. Diese kann, wie der Name selbst, frei gewählt werden. Besser ist es allerdings, die ungeschriebenen Gesetze der Einhaltung bestimmter genormter Dateikennungen einzuhalten. Der Vorteil dieser inoffiziellen Norm ist, daß der Benutzer an dieser Dateikennung bereits sieht, um was für eine Art Datei es sich handelt.

In Abbildung 3 ist eine Auflistung einer Reihe von Zusätzen einzusehen, die

Ein typisches Inhaltsverzeichnis einer Festplatte (ohne Unterdirektories)

Volume in Laufwerk C hat keinen Namen
Verzeichnis von C:\

AUTOEXEC	BAT	177	11-13-89	1:40p
COMMAND	COM	24044	1-01-80	1:13a
CONFIG	SYS	95	11-13-89	1:03p
INST	EXE	34896	1-01-80	1:29a
KEYBGR	EXE	3029	2-09-87	6:19p
LIST	EXE	7824	7-09-89	9:23a
SPIEGEL	FIL	50688	11-13-89	7:36p

7 Datei(en) 8724480 Bytes frei

Abbildung 2:

Dateikennungen

APP	Application-Datei - GEM
ASC	ASCII-Datei
ASM	Assembler-Quellprogramm
BAK	Sicherheitskopie, die beim Überschreiben von Dateien aller Art erstellt werden
BAS	BASIC-Quellprogramm
BAT	Batchdatei
BIN	Binärdatei
C	C-Quellprogramm
COM	Befehlsdatei
DOC	Datei, die Erklärungen zu anderen Programmen liefert
EXE	Befehlsdatei
GEM	GEM-Datei
HEX	Datei hexadezimalen Inhalts
LIB	Bibliotheksdatei
OBJ	Objektdatei
PAS	Pascal-Quellprogramm
SIK	Word-Sicherheitskopie
SYS	Systemdatei
TMP	temporäre Datei
TXT	Textdatei

Abbildung 3:

gerade für den Einsteiger viele unbekannte Begriffe enthält. Sie sollen hier unkommentiert bleiben, da sich die meisten im Laufe unserer Reihe selbst erklären werden.

Weitere Angaben in unserem Inhaltsverzeichnis sind Informationen über die Länge der Datei sowie das Datum und die Zeit, an dem sie erstellt bzw. zum letzten Mal verändert wurde.

Der Umgang mit solchen Dateien ist relativ einfach. Ein Blick auf die Dateikennung informiert uns wie beschrieben über die Dateiart. Lautet diese Endung .EXE, .COM oder .BAT, so handelt es sich um ausführbare Programme. Diese lassen sich meist problemlos durch simple Eingabe des Dateinamens (ohne Punkt und Kennung) zur Ausführung bringen. Alle anderen Dateien sind nicht ausführbar, es sei denn, ein findiger Programmierer arbeitet mit ebenso listigen Tricks. Wir aber gehen vom Normalfall aus.

Mit Hilfe des Type-Befehls ist der Benutzer in der Lage, den Inhalt aller Dateien auf den Bildschirm zu bringen, was sich jedoch nur im Fall von Text- oder Hochsprachenquelldateien empfiehlt. Bei den anderen führt dieser Befehl lediglich dazu, daß eine Anhäufung scheinbar wirrer Zeichen auf dem Monitor entsteht, und der Computer einige Male an zu piepsen fängt.

Editieren einer Textzeile

Wenn wir einen Befehl unter MS-DOS eingeben, so bemerken wir spätestens beim ersten Vertipper, daß ein Editieren, also eine Korrektur des Fehlers, nicht ohne Probleme durchzuführen ist. Meist ist es einfacher, die Zeile komplett neu zu schreiben. Der Grund dafür liegt darin, daß MS-DOS mit einem Zeileneditor arbeitet. Der Befehlsinterpreter erkennt eine bestimmte Anzahl von eingegebenen Zeichen. Er beginnt mit der Analyse des Eintrags, nachdem der Benutzer durch Druck auf die Ausführungstaste (RETURN) die Abarbeitung befiehlt. Da ein freies Positionieren des Cursors nicht unterstützt wird, ist man vor allem bei einem Tippfehler am Anfang des Befehls gezwungen, je nach Länge des Befehls mitunter viel Schreibarbeit zu leisten.

Ist der Befehl erst einmal durch Return bestätigt, so besteht allerdings in gewissen Grenzen eine Editiermöglichkeit. MS-DOS schreibt eine gerade eingegebene Befehlszeile in einen Zwischenspeicher, den sogenannten Puffer. Die Erinnerung an diesen Befehl behält MS-DOS solange im Gedächtnis, bis ein neuer eingegeben und mit RETURN bestätigt wird.

Solange sich ein Befehl im Puffer befindet, stehen dem Benutzer diverse Funktionen zum Editieren des Pufferinhaltes zur Verfügung. (Eine Übersicht finden Sie in Abbildung 4).

Spielen Sie mit diesen Befehlen. Probieren Sie alle denkbaren Kombinationen, und lernen Sie so den Umgang mit diesen Editierbefehlen. Ignorieren Sie bei diesem 'Spiel' alle Fehlermeldungen, diese haben in diesem Fall keinerlei Bedeutung. Tippen Sie beispielsweise folgende Zeile ein:

BEFEHL TEST1 TEST2 <RETURN>

Drücken Sie nun die F1-Taste, dann sehen Sie, wie der erste Buchstabe aus unserer ursprünglichen Befehlszeile auf dem Monitor erscheint. Weitere Drücke auf diese Taste holen die nächsten Zeichen aus dem Puffer auf den Bildschirm. Sie können natürlich zwischendurch andere Zeichen von der Tastatur aus einfügen oder überschreiben. Zum Umschalten zwischen diesen beiden Modi dient die <INS>-Taste.

Die F2-Taste dient zum Kopieren von Buchstaben und Worten bis zum Auftreten eines von uns gewählten Zeichens. Um in unserem Beispiel zu bleiben: Nach Eingabe unserer Zeile würde ein gleichzeitiges Drücken der F2-Taste und der '1' bewirken, daß alle Zeichen bis zur 1 auf dem Monitor erscheinen. Im Gegensatz dazu bewirkt gleichzeitiges Drücken von F4 und '1', daß alle Zeichen bis zur 1 im Puffer übersprungen werden. F3 holt den gesamten Restinhalt des Puffers ohne "Wenn und Aber" auf den Monitor, F5 kopiert die momentane Befehlszeile in den Puffer und löscht den dort befindlichen Inhalt. Die Taste F6 haben wir bereits benutzt. Wir erinnern uns, daß damit die Eingabe unserer Batchdatei beendet wurde. F6 steht für die Tastenkombination CTRL-Z (CTRL-Z zeigt dem Betriebssystem an, daß eine Datei ihr Ende erreicht hat).

Lassen Sie uns an einem weiteren konkreten Beispiel die Funktionsweise der Editiermöglichkeiten darlegen. Sie wollen die Textverarbeitung Context starten und vertippen sich bei der Eingabe:

CONEXT

Durch gleichzeitiges Drücken von F2 und 'e' erscheint 'CON' hinter dem Prompt. Der Editor befindet sich in der Grundeinstellung im Überschreibmodus, wir aber wollen zwecks Korrektur der Zeile ein Zeichen (t) einfügen. Wir schalten also um. <INS> bringt uns in den gewünschten Modus. Nun geben wir den einzufügenden Buchstaben ein. Ein erneuter Druck auf die <INS>-Taste schaltet wieder zurück in den Überschreibmodus. Als letztes gilt es, den Rest des Puffers auszulesen, F3. Die korrigierte Eingabe befindet sich nun startbereit zur Ausführung.

Viele werden sagen, daß es einfacher sei, gleich den neuen Befehl korrekt einzugeben, da diese Methode zu umständlich erscheine. Das ist aber nur am Anfang der Computerlaufbahn so. Mit der Zeit gehen Ihnen diese Befehle in Fleisch und Blut über und sorgen so dafür, daß Ihnen in Zukunft einiges an Tipparbeit erspart bleibt.

Ausblick

In den nächsten Ausgaben beschäftigen wir uns mit so gängigen Ausdrücken wie Booten. Wir befassen uns mit der Einrichtung einer auf Ihre Bedürfnisse abgestellte CONFIG.SYS-Datei und erklären, wie man sich mittels einer AUTOEXEC.BAT-Datei erhebliche Arbeit sparen kann.

(jf)

Editierbefehlsverzeichnis

F1	– Ein Zeichen aus dem Puffer kopieren
F2 +	– Kopieren der Zeichen aus dem Puffer, bis das für + stehende Zeichen auftritt
F3	– Kopieren der Pufferrestes
F4 +	– Überspringen aller Zeichen bis das für + stehende Zeichen auftritt
F5	– Eingabezeile in den Puffer kopieren
F6	– Entspricht CTRL-Z, was ein Dateiende markiert
INS	– Schaltet zwischen Eingabe- und Überschreibmodus hin und her
DEL	– Zeichen aus dem Puffer wird gelöscht
ESC	– Eingabe wird ignoriert Eine neue Eingabe kann begonnen werden

Abbildung 4:

Superpack

ausverkauft

CPC Sonderhefte Nr. 3, 5, 6, 7

4 Stück = 28,– DM

Jetzt zugreifen

Halber Preis/ganze Leistung

Jetzt wird es aber höchste Eisenbahn, daß Sie endlich unsere CPC-Sonderhefte kennenlernen!
Kraftfutter für Ihren CPC - jetzt zum günstigen Preis.

NEU CPC-Sonderheft 8

Programme

Bericht
DFÜ- Ihr Anschluß an die Zukunft.
Alles über das Thema DFÜ. Von Zubehör bis hin zu Telefonnummern.
Inklusive Protokolle.

Anwendung
Die Weltzeit im CPC
Berechnungen von Zeitzonen in aller Welt.

Blick zu den Sternen
Ein Satellitenberechnungsprogramm.

Spiele
Golddukaten und Schiffskanonen.
Super-Grafik-Handelssimulation à la Hanse.

Ran an die Kisten
Super-Adaption des Strategiespiels "Sokoban". Mit eigenem Editor!

Aus dem Inhalt

Bistro "La Cuisine"
Steuern Sie einen programmierbaren Roboter, der hoch hinaus will.

Tips und Tricks
Nützliche Betriebsadressen im CPC.

Klein, aber oho
Viele Super-Programme in 1kByte.
Schnell abzutippen, mit überraschender Wirkung.

Hardware
Das CPC Fitness-Center.
Der absolute Wahnsinn. Eine Joystick-Matte im Selbstbau. Ein Vergnügen für die ganze Familie.

CP/M intern
Alles über das Betriebssystem CP/M.



DMV-Sammelordner
2 Stck. 15,80 DM*

nur 14,– DM*

* Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.





Biete Software

Astrologie mit Computer
International geschätzte Astrologenprogramme, professionelle Deutungsprogramme, Lernprogramme für Anfänger, Handschriftanalyse, Bio-Rhythmus, Astro-I-Ging.
Info gegen DM 2,- in Marken.
Astron, K.W. Bonert, Peter-Marqu.Str.4a, 2000 Hamburg 60 G

Software für CPC 6128
Multiplan 99,–, dbase II 99,–,
M&T Finanzbuchhaltung 148,–
Datamat 78,–, Hi Soft Pascal 78,–
Devpac Assembler 78,–,
HOT SPACE COMPUTER CENTRUM
8330 Eggenfelden, Schellenbruckstr. 6
0 87 21/65 73 G

CPC-PD SPIELE UND ANWENDERPRG.
LISTE 100PF bei: Peter Breuker,
Rektenstr. 10, 4930 Detmold 1

CPC - SPIELE (DISK) ab 12,– DM nur bei EDV-CLOOTS G

Wirtschaftliche Programme
für die Arztpraxis auf
dem Schneider CPC, Joyce, PC
Fa. EFFEKTA, Am Wiggert 9c
45 Osnabrück, 05 41/44 24 16 G

■■■■■■■■■■■■■■■■■
Lohn- und Einkommensteuer 89:
STEUERMAT 89: Analyse aller Einkunftsarten, jährl. Aktu, Tabelle, Ausdruck aller Angaben auf Hauptantrag u. Anlagen N / KSO 98,- DM
Einnahmeüberschuß-Rechnung:
FIBUMAT: 142,- DM
FORMULARPRINT: 49,- DM
f. IBM-komp. PC/AT oder JOYCE
Demo: je 15,- DM, (3,5" und 3" - Disk: 5,- DM Aufpreis)
EDV-SYSTEME-FARIN, FF-Software, Neuenhauserstr. 76, 4460 Nordhorn
TEL 0 59 21/3 94 22, FAX 0 59 21/3 36 22
■■■■■■■■■■■■■■■■■ G

Dias ordnen mit Computer
CPC 464/664/6128, JOYCE und PC bis zu 100000 Dias; Suchzeit 1 Sekunde. Info gegen Rückporto bei: Dipl.-Ing. W. Grotkasten, Birnenweg 6, 7060 Schorndorf
Tel. 0 71 81/4 28 46 G

BONZO'S SUPER MEDDLER Vs. 4.0: DAS VIELSEITIGSTE PROGRAMM FÜR BAND-DISK-KOPIEN (alle CPC's). Auch für allerneuste Kopierschutzmethoden! 3"-Disk mit über 1000 (!!) Lösungshinweisen nur DM 65,- + Versandk. Ausführl. Infos gg. Freiumschlag! MARTINA HIPPCHEN, PF 100966, 5000 Köln 1 G

ACHTUNG !! ERWACHSENE
10 Disk. 5 1/4 oder 3 1/2 mit delikaten Programmen f. IBM PC/Kompatible
Nur DM 40,- VS/Bar. M. Karbach
Remscheider Str. 18, 5650 Solingen G

ENDLICH! PD-SOFTWARE NUR DM 1,
Für IBM/Kompat. Kat-Disk.gratis.
M. Karbach, Remscheider Str. 18
5650 Solingen 1 G

■■■■■■■■■■■■■■■■■
Software für JOYCE je 98 DM !
- FiBu (profess. Prog., DATEV-Konten)
- Finanzmathe: Zinsen, Annuitäten, Raten, Tilgungspläne (20 Prog.)
Kellmann, Wilhelmstr. 71, 44 Münster
■■■■■■■■■■■■■■■■■

■■■■■■■■■■■■■■■■■
PD-SERVICE-LAGE bietet Ihnen:
Public-Domain & Shareware z.B:
über **800 ! DEUTSCHE Programme!**
über **350 ! Disk PD-SPIELE** und
neuste INTERNAT. Programme !!!
Kopierkosten: 4,50-2,70 DM !!
Alleinvertrieb PC-TEXT 2.0 & PC-FAKT v. Roland Otter !
Kat. f. IBM/kompat. g. 1,80 Porto
PD-SERVICE-LAGE: Bernd Schulz
Hasselstr. 38, 4937 Lage/Lippe
Auch alles auf 3,5" lieferbar !
NEU bei uns: Low-Cost-Software !
■■■■■■■■■■■■■■■■■ G

CPC & JOYCE PUBLIC DOMAIN
Ein Info ist für 2x100 Pf erhältlich
PDI, Pf 1118, D-6464 Linsengericht G

CPC 6128: Statistik! 0 52 25/37 63

JOYCE: wg. Systemwechsel z. verk: GerdesMouse 110,-, dBase II 90,-, Turbo Pascal, TPToolbox, TPTutor zus. 230,-, Databox JOYCE-SH2, 3 je 25,-, LocoScript 2.16 frz. 100,-, Loco-Spell frz. 100,-, Spiele: Fairlit Aftershock, CyrusII je 20,- (zus. 50,-), Tel 05 61/40 38 70

Lohn- und Einkommensteuer 1989
Druckerausgabe + Datensicherung.
Ausführliche Anleitung. Info 2 DM
3"-Disk für CPC 79,- DM + VP.
Versand gegen Vorkasse oder NN.
89er-Aktualisierung 35,- DM.
S. Teurich, Mesternstraße 6,
4952 Porta Westfalica 3 G

VERK. ORIG. CPC-SOFT: STOLE A MILLION, GREYFELL, PALITRON, EIDOLON, CHOLO, ROCKY HORR. SHOW, SUPERCOPY, FLOPPY FD-1.
TEL 0 91 01 27 87

Biete Hardware

CPC464/Grün + DD1 + Drucker + Bücher + Software VB 850,-, TEL 0 97 08/67 38

Gebrauchtcomputer mit Garantie
SCHNEIDER - AMSTRAD - PEACOCK
CPC's * JOYCE * PC * XT * AT *
HARDWARE **************
SOFTWARE * ab sofort noch *
BÜCHER * günstiger *
ZEITSCHRIFTEN **************
Ersatzteile & Reparaturdienst
Alles zu echten Superpreisen.
Ständiger Ankauf
Höchstpreise für Ihren "ALTEN" bei Computer - NEUKAUF !!!!
Sofort unseren KATALOG anford.
Schutzgebühr 3,- DM in Marken oder einfach anrufen!
EDV-Cloots
5132 Übach, Palenberg 5
Zeisstr. 7, Tel.: 0 24 51/4 66 08 G

JOYCE 8512 + LOCO 2.16 + WORDSTAR 3.0 + par./ser. Schnittstelle CPS 8256 + div. Softw. u. Zubehör NP 2300,- VB 1000, Tel. 0 64 03/6 74 46

Verk. 5 Fischertechnikkästen + 3 Motoren + Batterienpacks + Sort. Kasten. Preis VB. T. 0 71 53/3 24 28

CPC664 FARBE + DIV. SOFTWARE VB. DM 700. TEL. 0 40/6 72 91 19

Joyce Plus, CPS 8256, GERDES MOUSE-PACK, CA. 700KB ORIGINAL SOFTWARE AUF 70 DISKETTEN. PR. VS, TEL 0 58 38-12 52

BTX-MODUL FÜR 6128 DM 250. KPL. TEL. 0 89/49 62 26

464er COLOR + 2 x JOY + SOFTW. 495, ERW. 256KB = 95,-, DDI-1 = 195, NLQ-401 = 195,-, MIRAGE-IMAG. 45, WORDST. DBASE MULTIPL. je 75, Orig. SPIELE 10,-, HEFTE 1, Bücher TEL. 07 61-48 23 35

CPC6128, GRÜN, MULTIPLAN, DR. DRAW, WORDSTAR, DBASEII, 20 DISKETTEN, SPIELE, 2 BÜCHER (BASIC, CPM) ZUB. TOPZUST., 500 DM, 0 71 51/6 43 74

VERKAUFE DATAPHON S210-2 OHNE ANLEITUNG + KABEL 9 POLSUB-0 AUF 25 POL, PREIS 220,-, PER NACHNA., CHIFFRE 89/00110PC

Geprüfte Gebrauchtgeräte mit Garantie. Neue und gebr. Schneider und Amstrad PC, Floppy/Festplatten/ PC-MM und Farbmonitor. Neue und gebr. CPC/Floppy/Drucker. Ankauf bei Systemwechsel, Reparaturservice, Manfred Kobusch, Bergenkamp 8, 4750 Unna, 0 23 03/1 33 45 G

CPC464 + Floppy + Drucker viel Literatur, ca. 100 Spiele, Systembuch, 25 Disketten 900 DM, 0 26 24/29 59

Vortex F1-XRS für CPC 6128 für DM 450,- VHB, Tel.: 0 61 51/6 37 60

Ich verkaufe PC 1640 m. 2. Disk. 5,25", Festpl. 20MB u. Monochrom Monitor für 2500,- DM, Tel: 0 61 87/84 27 Chiffre 89/00113 PC

Floppy 5 1/4 211,-/Autostart f. CPC 6128, 664 (Strom einschalten, Programm drin) 100,-/Sprechi ab 99,-/Kondensator Meßgerät 1/100 (!) pF genau 99,-/Modulator (Videoeingang) 119,-/CNC Stellinger Weg 43/5, 2000 Hamburg 20. Computer bitte angeben G

PCW8512, Vortex-HD 60MB, Zubehör, Disketten, Software, Literatur, VHB DM 2500,-. Werner Leonhardt, Hebbelstr. 1, 2351 Trappenkamp

CPC 6128 (grün) + Seikosha sp 180 AI + 30 Disks + Comp. P. (v. SM) ab 10/87, DM 700,-, 0 50 21/55 96

AMSTRAD PC 1640 HD20 MD + DMP3160 + Co. Proz. + 40 Disk. zus. VB 2250 DM. HP 41CX + Cardreader + Mathe-Mod + Finanz-Mod + Drucker + Mag. karten, zus. VB 800 DM, Michael Rödder, Tel. 0 22 43-29 31

Lichtgriffel nur DM 49,-, Versand gegen Scheck/Nachnahme. Info gratis! Computer angeben! Anschluß an jeden (!) Computer möglich. Standardversion für Amstrad lieferbar. Firma Klaus Schißlbauer, Postfach 1171R, 8458 Sulzbach, Tel.: 0 96 61/65 92 oder 09 41/99 99 15 bis 21 Uhr G

Verkaufe PC 1640/HD20 MONOCHROME-MONITOR 10 MON. ALT MIT ORIG. HANDBUCH + SOFTWARE, TEL.: 0 21 06/4 60 01, DORMAGEN, VB DM 1900,-

VORTEX RS-232-C Interface mit ROM für CPC 664 + Softw. Disk 3", DM 100,-, 0 93 52/72 33 ab 19 h

Suche Software

Suche/Spiele/u. Programme/für Joyce PCW8512/M. Florschütz/Tel. 0 95 64/49 70

STOP-PRESS + STOP PRESS-UTILITIES gesucht. Tel.: 07 31/38 32 20

Suche Hardware

Aufsteiger kaufen Ihren "Neuen" nur bei EDV-CLOOTS G

Orig. AMSTRAD RS232: 02 21-21 53 02

Verschiedenes

■■ Über 555 POKEs & Codes ■■
+ 1 POKE-Sucher für CPC-Spiele.
NUR 5 DM (Schein, 10er geht auch)
Abeln, Danziger 8a, 2841 STEINFELD

upLink bringt Daten u. Progr. vom CPC u. JOYCE (m. CPS 8256) direkt zum PC/XT/AT. Transfer über die parallelen Schnittstellen. Komplett 198 DM - NN. Info. / Bestellung: CONCEPTION GmbH, Hubertusweg 14, 2000 Hamburg 61, Telefon: 0 40/58 45 03
Telefax 0 40/58 73 52 G

AMSTRAD PC 1512/1640 USER-CLUB bietet mtl. Zeitschrift, Software, Hilfe und mehr für alle.
Inf. gg. Rückp. von Rolf Knorre, Postf. 20 01 02, 5600 Wuppertal 2

Suche Kontakt zu CPC-Usern:
Ronald Kunz, Ricarda-Huch-Str. 18
DDR-9400 Aue.

VERKAUFE CPC 6128 + DMP 2000 + GRUENMONITOR + 50 SPIELE + LEERDISKETTEN + DATASETTE + 2 JOYSTICKS, VB 1000 DM, TEL. a 17 Uhr 0 22 02/8 53 41

CPC 6128 m. viel Hard u. Software Tel. 0 73 07/54 85

Club

USER-CLUB für JOYCE-Anwender! Info gegen Rückporto von: JOYCE-USER-CLUB c/o Jc. Berghof, Roseggerstr. 5, 5600 Wuppertal 2

Berlin

Castrop-Rauxel

Löhne/Ostwestfalen

Nürnberg

Soltau

Basel

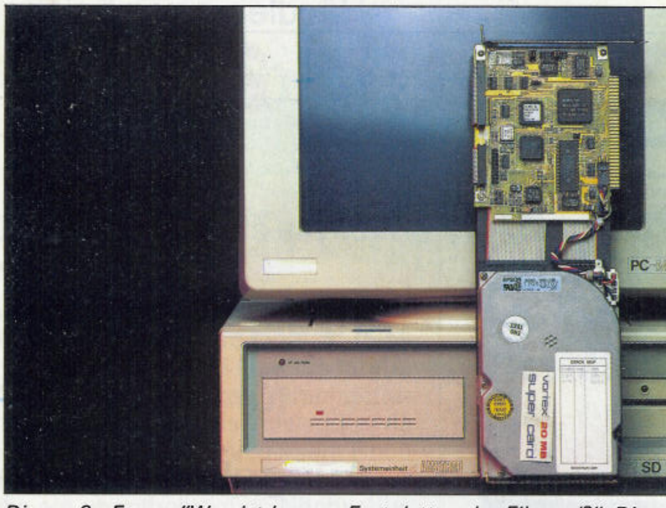
Anzeigenschluß für die Ausgabe 3/90 von PC International ist der 12.1.90

Erscheinungstermin ist der 28.2.90

Eintragungen im Händlerverzeichnis, nach Städten geordnet, kosten je mm Höhe 6,– DM bei einer Spaltenbreite von 58 mm.

Einträge möglich mindestens 6 × innerhalb eines Insertionsjahres.

Nähere Informationen:
DMV-Verlag
Wolfgang Brill
Telefon (05651) 809-380



Die große Frage: "Was ist besser: Festplatte oder Filecard?". Dieses Problem wird in der nächsten Ausgabe erörtert

»PC International« 2/90 erhalten Sie ab:

JANUAR
31
MITTWOCH

CPC-PROGRAMME:

Lange und kalte Winterabende verleiten den Computerbesitzer, noch öfter mit seinem "Liebling" zusammenzusitzen, als in warmen Tagen. Folglich bleiben auch Probleme mit den übrigen Familienmitgliedern nicht aus. Als Abhilfe schlagen wir vor: Laden Sie Ihre Wohnungsgenossen mit zu einer Partie "Slysprite" ein. In diesem Kombinations- und Taktikspiel werden so manche Schlachten geschlagen werden.



Kombination ist bei "Slysprite" angesagt, zwei Spieler geben sich ein spannendes Duell

Für diejenigen, die sich keine Zeit für Spiele nehmen können, werden sich ernsthaftere Programme zu Gemüte führen. Für diesen Anwenderkreis bietet sich "Rechnomat" an. Mit diesem Programm lassen sich Rechnungen auf einfache Weise erstellen. Damit Adressen nicht immer neu eingegeben werden müssen, werden auch diese mit verwaltet. Ein Muß für jeden, der viele Rechnungen zu schreiben hat.

Tips und Tricks auf dem CPC:
Wir haben für Sie auf einigen Seiten interessante POKE- und CALL-Befehle rund um den CPC zusammengestellt.

Weiterhin erwartet Sie eine besondere Art der Bildschirmmanipulation. Weitere Tips zu Multiplan und dem LQ-3500-Drucker ergänzen die Rubrik.

PCW:

Für alle, die noch keinen Kalender oder Terminplaner für das neue Jahr besitzen, haben wir eine Spezialität. Drucken Sie sich Ihren Kalender mit allen für Sie wichtigen Daten selbst aus.

Für das Spiel "Head Over Heals" haben wir ein kleines Patchprogramm, welches es unter anderem ermöglicht, eine Autostartdiskette zu erstellen.

Der Video-Display-Editor aus der englischen Public Domain im Test.

Wenn Sie eine Analoguhr in BASIC einbauen wollen und die BASIC-Erweiterung von Gerdes besitzen, können Sie unsere Routine benutzen.

PC:

Festplatte oder Filecard? Vor dieser Frage steht jeder, der seinen Amstrad PC aufrüsten will. Was bei der Wahl beachtet werden sollte und wie der Einbau vonstatten geht, lesen Sie in Ausgabe 2/90.

Heroes Quest, das erste Sierra-Abenteuer mit starken Rollenspieleinflüssen, ist erschienen. Kann es die Erwartungen befriedigen, die ein neues Sierra-Spiel automatisch an sich stellt? Wir haben es für Sie unter die Lupe genommen.



Heroes Quest ist da. Sierra-Online präsentiert sein neuestes Software-Produkt

Zwei weitere Möglichkeiten, Ihre Daten vor fremden Zugriffen zu schützen, werden wir Ihnen in der nächsten Ausgabe nicht vorenthalten. Die eine ist ein Programm namens Savedir, die andere ein kleiner Zugriff in der COMMAND.COM Datei.

In unserem zweiten Teil der Reihe MS-DOS, »Die ersten Berührungen«, befassen wir uns unter anderem mit der richtigen Einrichtung der Dateien AUTOEXEC.BAT und CONFIG.SYS.

DIE INSERENTEN

SOFTWARE EXPERIMENT

Autor: Matthias Uphoff

Von der Idee zum Programm

Nehmen Sie teil an dem Abenteuer "Computer"! In zehn lebendig und leicht verständlich geschilderten Reisen in die Welt der Computergrafik, Simulation und künstlichen Intelligenz erarbeitet der Autor mit Ihnen zusammen insgesamt 17 interessante Programme, die Sie auf Diskette/Kassette erhalten.

Das umfangreiche, 180 seitige Handbuch vermittelt Ihnen auf anschauliche und unterhaltsame Weise, wie aus Algorithmen Datenstrukturen und letztendlich Software entsteht: Von der Idee zum Programm - der ideale Kursus zum Einstieg in das Computerwissen, für Hobby und Schulunterricht. Sehen Sie selbst, zu welchen phantastischen Grafiken und verblüffenden Intelligenzleistungen der CPC bei raffinierter Programmierung fähig ist!

Weiterhin:

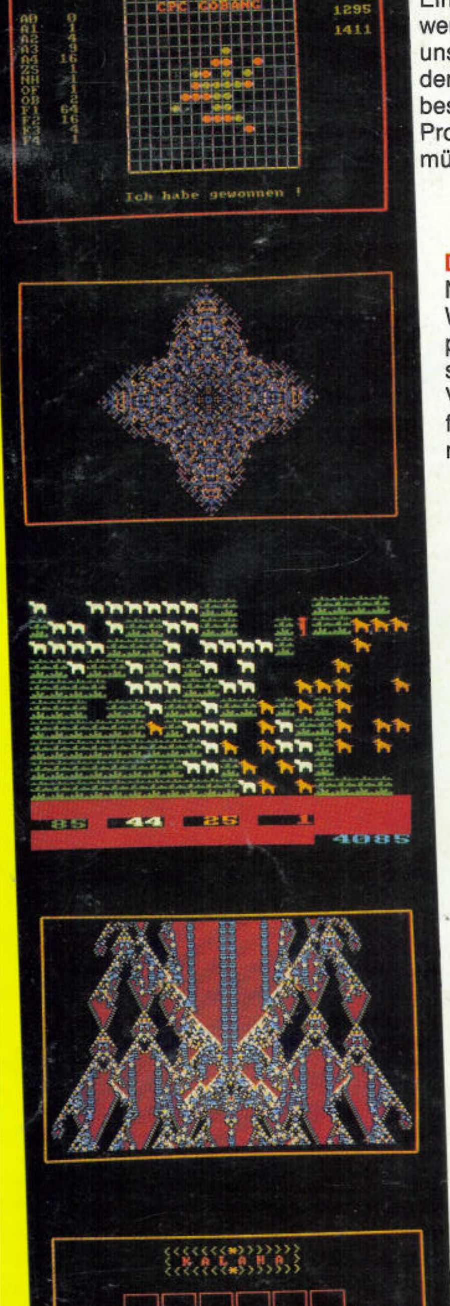
Komplexe Grafik
Sie lernen die mathematischen Grundlagen von Fraktalgrafiken und erarbeiten einen kompletten Fraktalgenerator zur Erzeugung von "Apfelmännchen"-Bildern.

Der Computer lernt
Erleben Sie ein Paradebeispiel für "Künstliche Intelligenz": Das Programm "Minischach" ist lernfähig und wird mit jeder Partie ein bißchen besser.

Wordmaster
Sie raten ein Wort, das sich der Computer ausgedacht hat. Dann rät der Rechner ein Wort, das Sie sich ausgedacht haben. Wer braucht weniger Versuche? Sie werden sich wundern...

Wordketten
Mit einem raffinierten Algorithmus löst der CPC Wortketten-Rätsel. Oder er erfindet neue Rätsel - ganz wie Sie wollen.

Entwicklungshilfe
Eine gut ausgestaltete Simulation versetzt Sie in die Lage eines Entwicklungshelfers, der eine Hungersnot abzuwenden versucht. Aber Ihre Mittel sind begrenzt.



für CPC 464/664/6128

Der Computer als Stratege
Ein raffiniertes System von Spielzugbewertungen macht den CPC zu einem fast unschlagbaren Gegner im Gobang-Spiel, der Ihnen anfangs saftige Niederlagen bescheren wird. Keine Angst - das Programm ist zu schlagen - aber wie, das müssen Sie selbst herausfinden.

Das Spiel des Lebens
Nach natürlichen Prinzipien des Wachstums erzeugt der Computer phantastische, kristallförmige Farbstrukturen. Ändern Sie die Regeln und Voraussetzungen und sehen Sie, was für neue, noch nie dagewesene Formen sich ergeben.

Das Ökologie-Experiment
Der Computer zeigt Ihnen in animierter Farbgrafik einen simulierten Lebensraum mit Weideland, Schafen und Wölfen. Erforschen Sie die ökologischen Gesetzmäßigkeiten, und schaffen Sie ein stabiles, natürliches Gleichgewicht.

Pascal läßt grüßen
Ein erstaunliches Programm enthüllt verborgene Strukturen im sogenannten "Pascalschen Dreieck". Die grafische Darstellung ergibt traumhaft schöne und farbige Musterbilder.

Mit roher Rechengewalt
Mit dem "Brute-Force"-Algorithmus darf der CPC so richtig loslegen und spielt mit roher Rechengewalt in dem afrikanischen Strategiespiel "Kalaha" alle Gegner an die Wand - oder können **Sie** ihn trotzdem schlagen?

Das Software-Experiment

- Siebzehn verschiedene, lauffähige Programme
- 180 Seiten Anleitung

Kassette nur DM 29,– *
Diskette nur DM 39,– *

* Unverbindliche Preisempfehlung. Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4.- DM bzw. für das Ausland 6.- DM Porto und Verpackung.

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege

